



WAGO-I/O-SYSTEM 750 **BACnet - Objekte, Eigenschaften, Dienste**

BACnet/IP-Controller 750-830

Version 1.0.2, gültig ab HW/SW-Version 03/01.04.27(03)

Impressum

© 2010 by WAGO Kontakttechnik GmbH & Co. KG
Alle Rechte vorbehalten.

Inhalte dieser Dokumentation sind teilweise dem BACnet-Standard 135-2004 entnommen oder basieren auf dem Originalinhalt. Diese Inhalte sind urheberrechtlich geschützt.

Für diese Inhalte gilt: ©2004, American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. (www.ashrae.org). Reprinted by permission from 2004 ASHRAE Standard-135. This material may not be copied nor distributed in either paper or digital form without ASHRAE's permission.

Für das BACnet-Logo gilt: BACnet[®] is a registered trademark of ASHRAE.

WAGO Kontakttechnik GmbH & Co. KG

Hansastraße 27
D-32423 Minden

Tel.: +49 (0) 571/8 87 – 0
Fax: +49 (0) 571/8 87 – 1 69

E-Mail: info@wago.com

Web: <http://www.wago.com>

Technischer Support

Tel.: +49 (0) 571/8 87 – 7 77
Fax: +49 (0) 571/8 87 – 87 77

E-Mail: tcba@wago.com

Es wurden alle erdenklichen Maßnahmen getroffen, um die Richtigkeit und Vollständigkeit der vorliegenden Dokumentation zu gewährleisten. Da sich Fehler, trotz aller Sorgfalt, nie vollständig vermeiden lassen, sind wir für Hinweise und Anregungen jederzeit dankbar.

E-Mail: documentation@wago.com

Wir weisen darauf hin, dass die im Handbuch verwendeten Soft- und Hardware-Bezeichnungen und Markennamen der jeweiligen Firmen im Allgemeinen einem Warenzeichenschutz, Markenzeichenschutz oder patentrechtlichem Schutz unterliegen.

Inhaltsverzeichnis

Impressum.....	2
Inhaltsverzeichnis.....	3
1 Hinweise zu dieser Dokumentation	11
1.1 Gültigkeitsbereich	11
1.2 Urheberschutz	11
1.3 Symbole.....	12
1.4 Darstellung der Zahlensysteme	13
1.5 Schriftkonventionen	13
2 Wichtige Erläuterungen	14
2.1 Rechtliche Grundlagen.....	14
2.1.1 Änderungsvorbehalt	14
2.1.2 Personalqualifikation.....	14
2.1.3 Bestimmungsgemäße Verwendung der Serie 750.....	14
2.1.4 Technischer Zustand der Geräte.....	15
2.2 Sicherheitshinweise.....	16
3 Programmierung des BACnet/IP-Controllers 750-830	18
4 Objekte im nicht-nativen Betrieb	19
4.1 Accumulator-Objekt.....	19
4.2 Analog-Value-Objekt.....	21
4.3 Averaging-Objekt.....	22
4.4 Binary-Value-Objekt.....	23
4.5 Calendar-Objekt	24
4.6 Command-Objekt.....	25
4.7 File-Objekt	27
4.8 Group-Objekt	28
4.9 Life-Safety-Point-Objekt.....	29
4.10 Life-Safety-Zone-Objekt.....	32
4.11 Multi-State-Value-Objekt	35
4.12 Notification-Class-Objekt	36
4.13 Pulse-Converter-Objekt.....	38
4.14 Schedule-Objekt.....	40
4.15 Trend-Log-Objekt	41
5 BACnet-Dienste.....	44
5.1 Alarm- und Ereignisdienste.....	44
5.1.1 Melden bei Wertänderungen (Change Of Value / State Reporting).....	46
5.1.2 Objektinternes Melden	49
5.1.3 Auftreten und Benachrichtigung von Alarmen und Ereignissen.....	54
5.1.3.1 Prioritätsklassifizierung von Alarmen und Ereignissen	56
5.1.4 Terminologie der Darstellung.....	57
5.1.5 AcknowledgeAlarm-Dienst.....	59
5.1.5.1 Struktur	59
5.1.5.2 Vorgehensweise des Dienstes.....	59
5.1.6 ConfirmedCOVNotification-Dienst	60

5.1.6.1	Struktur	60
5.1.6.2	Vorgehensweise des Dienstes	60
5.1.7	UnconfirmedCOVNotification-Dienst	61
5.1.7.1	Struktur	61
5.1.7.2	Vorgehensweise des Dienstes	61
5.1.8	ConfirmedEventNotification-Dienst	62
5.1.8.1	Struktur	62
5.1.8.2	Vorgehensweise des Dienstes	62
5.1.9	UnconfirmedEventNotification-Dienst	63
5.1.9.1	Struktur	63
5.1.9.2	Vorgehensweise des Dienstes	63
5.1.10	GetAlarmSummary-Dienst	64
5.1.10.1	Struktur	64
5.1.10.2	Vorgehensweise des Dienstes	64
5.1.11	GetEnrollmentSummary-Dienst	65
5.1.11.1	Struktur	65
5.1.11.2	Vorgehensweise des Dienstes	65
5.1.12	GetEventInformation-Dienst	66
5.1.12.1	Struktur	66
5.1.12.2	Vorgehensweise des Dienstes	67
5.1.13	LifeSafetyOperation-Dienst	68
5.1.13.1	Struktur	68
5.1.13.2	Vorgehensweise des Dienstes	68
5.1.14	SubscribeCOV-Dienst	69
5.1.14.1	Struktur	69
5.1.14.2	Vorgehensweise des Dienstes	70
5.1.15	SubscribeCOVProperty-Dienst	71
5.1.15.1	Struktur	71
5.1.15.2	Vorgehensweise des Dienstes	72
5.2	File-Access-Dienste	73
5.2.1	AtomicReadFile-Dienst	73
5.2.1.1	Struktur	73
5.2.1.2	Vorgehensweise des Dienstes	74
5.2.2	AtomicWriteFile-Dienst	75
5.2.2.1	Struktur	75
5.2.2.2	Vorgehensweise des Dienstes	75
5.3	Object-Access-Dienste	76
5.3.1	AddListElement-Dienste	76
5.3.1.1	Struktur	76
5.3.1.2	Vorgehensweise des Dienstes	76
5.3.2	RemoveListElement-Dienst	77
5.3.2.1	Struktur	77
5.3.2.2	Vorgehensweise des Dienstes	77
5.3.3	ReadProperty-Dienst	78
5.3.3.1	Struktur	78
5.3.3.2	Vorgehensweise des Dienstes	78
5.3.4	ReadPropertyConditional-Dienst	79
5.3.4.1	Struktur	79
5.3.4.2	Vorgehensweise des Dienstes	79
5.3.4.3	Parameter des Dienstes ReadPropertyConditional	80

5.3.4.3.1	Object Selection Criteria	80
5.3.4.3.2	Read Access Result	80
5.3.5	ReadPropertyMultiple-Dienst	80
5.3.5.1	Struktur	81
5.3.5.2	Vorgehensweise des Dienstes	81
5.3.5.3	Parameter des Dienstes ReadPropertyMultiple	81
5.3.5.3.1	Read Access Specification	82
5.3.5.3.2	Read Access Result	82
5.3.6	ReadRange-Dienst	83
5.3.6.1	Struktur	83
5.3.6.1.1.1	Range	83
5.3.6.2	Vorgehensweise des Dienstes	84
5.3.7	WriteProperty-Dienst	86
5.3.7.1	Struktur	86
5.3.7.2	Vorgehensweise des Dienstes	86
5.3.8	WritePropertyMultiple-Dienst	87
5.3.8.1	Struktur	87
5.3.8.2	Vorgehensweise des Dienstes	87
5.3.8.3	Parameter des Dienstes WritePropertyMultiple	88
5.4	Remote-Device-Management-Dienste	89
5.4.1	DeviceCommunicationControl-Dienst	89
5.4.1.1	Struktur	89
5.4.1.2	Vorgehensweise des Dienstes	89
5.4.2	ReinitializeDevice-Dienst	90
5.4.2.1	Struktur	90
5.4.2.2	Vorgehensweise des Dienstes	90
5.4.3	TimeSynchronization-Dienst	91
5.4.3.1	Struktur	91
5.4.3.2	Vorgehensweise des Dienstes	91
5.4.4	UTCTimeSynchronization-Dienst	92
5.4.4.1	Struktur	92
5.4.4.2	Vorgehensweise des Dienstes	92
5.4.5	Dienste Who-Has und I-Have	93
5.4.5.1	Struktur des Dienstes Who-Has	93
5.4.5.2	Vorgehensweise des Dienstes Who-Has	93
5.4.5.3	Struktur des Dienstes I-Have	94
5.4.5.4	Vorgehensweise des Dienstes I-Have	94
5.4.6	Dienste Who-Is und I-Am	95
5.4.6.1	Struktur des Dienstes Who-Is	95
5.4.6.2	Vorgehensweise des Dienstes Who-Is	95
5.4.6.3	Struktur des Dienstes I-Am	96
5.4.6.4	Vorgehensweise des Dienstes I-Am	96
6	Fehler-, Ablehnungs- und Abbruchcodes	97
6.1	Fehlerklasse – DEVICE	97
6.2	Fehlerklasse – OBJECT	98
6.3	Fehlerklasse – PROPERTY	99
6.4	Fehlerklasse – RESOURCES	100
6.5	Fehlerklasse – SECURITY	101
6.6	Fehlerklasse – SERVICES	102
6.7	Fehlerklasse - VT	103

6.8	Ablehnungsgründe	104
6.9	Abbruchgründe.....	105
7	Gesamtliste der BACnet-Objekt-Eigenschaften	106
7.1	Accepted_Modes.....	106
7.2	Ack_Required	107
7.3	Acked_Transitions	107
7.4	Action.....	112
7.5	Action_Text.....	114
7.6	Active_Text.....	114
7.7	Adjust_Value.....	115
7.8	Alarm_Value	115
7.9	Alarm_Values	116
7.10	All_Writes_Successful.....	119
7.11	Archive	119
7.12	Attempted_Samples	120
7.13	Average_Value.....	120
7.14	Buffer_Size	121
7.15	Change_Of_State_Count.....	121
7.16	Change_Of_State_Time.....	122
7.17	Client_COV_Increment	122
7.18	Count.....	123
7.19	Count_Before_Change.....	123
7.20	Count_Change_Time	124
7.21	COV_Increment	124
7.22	COV_Period.....	125
7.23	COV_Resubscription_Interval	125
7.24	Date_List.....	126
7.25	Deadband.....	127
7.26	Description	127
7.27	Device_Type	128
7.28	Direct_Reading	129
7.29	Effective_Period.....	129
7.30	Elapsed_Active_Time	130
7.31	Event_Enable	130
7.32	Event_State	132
7.33	Event_Time_Stamps	134
7.34	Exception_Schedule.....	135
7.35	Fault_Values	137
7.36	File_Access_Method.....	138
7.37	File_Size.....	138
7.38	File_Type	139
7.39	High_Limit.....	139
7.40	In_Process	140
7.41	Inactive_Text.....	140
7.42	Input_Reference	141
7.43	Last_Notify_Record.....	141
7.44	Life_Safety_Alarm_Values.....	142
7.45	Limit_Enable.....	143
7.46	Limit_Monitoring_Interval	143
7.47	List_Of_Group_Members	144

7.48	List_Of_Object_Property_References	145
7.49	Log_Buffer	146
7.50	Log_DeviceObjectProperty	147
7.51	Log_Enable	148
7.52	Log_Interval	148
7.53	Low_Limit	149
7.54	Maintenance_Required	149
7.55	Max_Pres_Value	150
7.56	Maximum_Value	150
7.57	Maximum_Value_Timestamp	150
7.58	Member_Of	151
7.59	Minimum_Off_Time	151
7.60	Minimum_On_Time	152
7.61	Minimum_Value	152
7.62	Minimum_Value_Timestamp	152
7.63	Mode	153
7.64	Modification_Date	153
7.65	Notification_Class	154
7.66	Notification_Threshold	154
7.67	Notify_Type	155
7.68	Number_Of_States	155
7.69	Object_Identifier	156
7.70	Object_Name	156
7.71	Object_Property_Reference	157
7.72	Object_Type	158
7.73	Operation_Expected	158
7.74	Out_Of_Service	159
7.75	Prescale	162
7.76	Present_Value	163
7.77	Priority	168
7.78	Priority_Array	168
7.79	Priority_For_Writing	169
7.80	Pulse_Rate	170
7.81	Read_Only	170
7.82	Recipient_List	171
7.83	Record_Count	172
7.84	Records_Since_Notification	173
7.85	Reliability	173
7.86	Relinquish_Default	176
7.87	Scale	177
7.88	Scale_Factor	177
7.89	Schedule_Default	178
7.90	Setting	178
7.91	Silenced	179
7.92	Start_Time	179
7.93	State_Text	180
7.94	Status_Flags	181
7.95	Stop_Time	183
7.96	Stop_When_Full	184
7.97	Time_Delay	184

7.98	Time_Of_Active_Time_Reset	186
7.99	Time_Of_State_Count_Reset	186
7.100	Total_Record_Count	186
7.101	Tracking_Value	187
7.102	Units	187
7.103	Update_Time	188
7.104	Valid_Samples	188
7.105	Value_Before_Change	189
7.106	Value_Change_Time	189
7.107	Value_Set	190
7.108	Weekly_Schedule	191
7.109	Window_Interval	192
7.110	Window_Samples	193
7.111	Zone_Members	193
8	Gesamtliste der BACnet-Dienst-Parameter	194
8.1	Acknowledged Transitions	194
8.2	Acknowledging Process Identifier	194
8.3	Acknowledgment Filter	195
8.4	Acknowledgment Source	195
8.5	AckRequired	195
8.6	Alarm State	196
8.7	Argument	196
8.8	By Position	196
8.9	By Sequence Number	197
8.10	By Time	197
8.11	Comparison Value	198
8.12	Count	200
8.13	COV Increment	200
8.14	Device Identifier	201
8.15	Device Instance Range High Limit	201
8.16	Device Instance Range Low Limit	202
8.17	Enable/Disable	203
8.18	End Of File	203
8.19	Enrollment Filter	204
8.20	Error Type	204
8.21	Event Enable	207
8.22	Event Object Identifier	207
8.23	Event Priorities	207
8.24	Event State	207
8.25	Event State Acknowledged	208
8.26	Event State Filter	208
8.27	Event Time Stamps	208
8.28	Event Type	209
8.29	Event Type Filter	209
8.30	Event Values	209
8.31	File Data	209
8.32	File Identifier	210
8.33	File Record Data	210
8.34	File Start Position	210
8.35	File Start Record	210

8.36	First Failed Element Number	211
8.37	First Failed Write Attempt	211
8.38	First Sequence Number	211
8.39	From State	212
8.40	I-Am Device Identifier	212
8.41	Initiating Device Identifier	212
8.42	Issue Confirmed Notifications	213
8.43	Item Count	213
8.44	Item Data	213
8.45	Last Received Object Identifier	214
8.46	Lifetime	214
8.47	List of Alarm Summaries	214
8.48	List of Elements	215
8.49	List of Enrollment Summaries	215
8.50	List of Event Summaries	215
8.51	List of Properties	216
8.52	List of Property References	216
8.53	List of Read Access Results	217
8.54	List of Read Access Specifications	218
8.55	List of Results	218
8.56	List of Selection Criteria	218
8.57	List of Values	219
8.58	List of VT Session Identifiers	219
8.59	List of Write Access Specifications	219
8.60	List of Results	220
8.61	Max APDU Length Accepted	220
8.62	Message Text	220
8.63	Monitored Object Identifier	221
8.64	Monitored Property Identifier	221
8.65	More Events	221
8.66	Notification-Class	222
8.67	Notification-Class Filter	222
8.68	Notify Type	223
8.69	Object Identifier	224
8.70	Object Name	226
8.71	Object Selection Criteria	226
8.72	Password	227
8.73	Priority	227
8.74	Priority Filter	228
8.75	Process Identifier	228
8.76	Property Access Error	228
8.77	Property Array Index	229
8.78	Property Identifier	230
8.79	Property Value	231
8.80	Range	232
8.81	Record Access	232
8.82	Record Count	233
8.83	Reference Index	233
8.84	Reference Sequence Number	233
8.85	Reference Time	234

8.86	Reinitialized State of Device.....	234
8.87	Relation Specifier.....	234
8.88	Request.....	235
8.89	Requested Octet Count.....	235
8.90	Requested Record Count.....	235
8.91	Requesting Process Identifier.....	236
8.92	Requesting Source.....	236
8.93	Result(-).....	236
8.94	Result(+).	237
8.95	Result Flags.....	238
8.96	Returned Record Count.....	238
8.97	Segmentation Supported	238
8.98	Selection Logic.....	239
8.99	Stream Access	239
8.100	Subscriber Process Identifier.....	240
8.101	Text Message Source Device	240
8.102	Time	240
8.103	Time Duration	241
8.104	Time Of Acknowledgment.....	241
8.105	Time Remaining.....	241
8.106	Time Stamp	242
8.107	To State	242
8.108	Vendor ID.....	242
8.109	Vendor Identifier.....	243
Abbildungsverzeichnis		244
Tabellenverzeichnis		245

1 Hinweise zu dieser Dokumentation

Hinweis



Dokumentation aufbewahren!

Diese Dokumentation ist Teil des Produkts. Bewahren Sie deshalb diese während der gesamten Lebensdauer des Gerätes auf. Geben Sie die Dokumentation an jeden nachfolgenden Besitzer oder Benutzer des Gerätes weiter. Stellen Sie darüber hinaus sicher, dass gegebenenfalls jede erhaltene Ergänzung in die Dokumentation mit aufgenommen wird.

1.1 Gültigkeitsbereich

Die vorliegende Dokumentation gilt für den BACnet/IP-Controller 750-830 und dient als Referenz für verwendete Objekte, Eigenschaften und Dienste.

Der BACnet/IP-Controller 750-830 darf nur nach Anweisungen der zugehörigen Betriebsanleitung und der Systembeschreibung zum WAGO-I/O-SYSTEM 750 installiert und betrieben werden.

ACHTUNG



Versorgungsauslegung des WAGO-I/O-SYSTEM 750 beachten!

Sie benötigen zusätzlich zu dieser Betriebsanleitung die Systembeschreibung zum WAGO-I/O-SYSTEM 750, die unter www.wago.com herunterzuladen ist. Dort erhalten Sie unter anderem wichtige Informationen zu Potenzialtrennung, Systemversorgung und Einspeisungsvorschriften.

1.2 Urheberschutz

Diese Dokumentation, einschließlich aller darin befindlichen Abbildungen, ist urheberrechtlich geschützt. Jede Weiterverwendung dieser Dokumentation, die von den urheberrechtlichen Bestimmungen abweicht, ist nicht gestattet. Die Reproduktion, Übersetzung in andere Sprachen sowie die elektronische und fototechnische Archivierung und Veränderung bedarf der schriftlichen Genehmigung der WAGO Kontakttechnik GmbH & Co. KG, Minden. Zuwiderhandlungen ziehen einen Schadenersatzanspruch nach sich.

1.3 Symbole

GEFAHR**Warnung vor Personenschäden!**

Kennzeichnet eine unmittelbare Gefährdung mit hohem Risiko, die Tod oder schwere Körperverletzung zur Folge haben wird, wenn sie nicht vermieden wird.

GEFAHR**Warnung vor Personenschäden durch elektrischen Strom!**

Kennzeichnet eine unmittelbare Gefährdung mit hohem Risiko, die Tod oder schwere Körperverletzung zur Folge haben wird, wenn sie nicht vermieden wird.

WARNUNG**Warnung vor Personenschäden!**

Kennzeichnet eine mögliche Gefährdung mit mittlerem Risiko, die Tod oder (schwere) Körperverletzung zur Folge haben kann, wenn sie nicht vermieden wird.

VORSICHT**Warnung vor Personenschäden!**

Kennzeichnet eine mögliche Gefährdung mit geringem Risiko, die leichte oder mittlere Körperverletzung zur Folge haben könnte, wenn sie nicht vermieden wird.

ACHTUNG**Warnung vor Sachschäden!**

Kennzeichnet eine mögliche Gefährdung, die Sachschaden zur Folge haben könnte, wenn sie nicht vermieden wird.

ESD**Warnung vor Sachschäden durch elektrostatische Aufladung!**

Kennzeichnet eine mögliche Gefährdung, die Sachschaden zur Folge haben könnte, wenn sie nicht vermieden wird.

Hinweis**Wichtiger Hinweis!**

Kennzeichnet eine mögliche Fehlfunktion, die aber keinen Sachschaden zur Folge hat, wenn sie nicht vermieden wird.

Information**Weitere Information**

Weist auf weitere Informationen hin, die kein wesentlicher Bestandteil dieser Dokumentation sind (z. B. Internet).

1.4 Darstellung der Zahlensysteme

Tabelle 1: Darstellungen der Zahlensysteme

Zahlensystem	Beispiel	Bemerkung
Dezimal	100	normale Schreibweise
Hexadezimal	0x64	C-Notation
Binär	'100' '0110.0100'	in Hochkomma, Nibble durch Punkt getrennt

1.5 Schriftkonventionen

Tabelle 2: Schriftkonventionen

Schriftart	Bedeutung
<i>kursiv</i>	Namen von Pfaden und Dateien werden kursiv dargestellt z. B.: <i>C:\Programme\WAGO-I/O-CHECK</i>
Menü	Menüpunkte werden fett dargestellt z. B.: Speichern
>	Ein Größer als Zeichen zwischen zwei Namen bedeutet die Auswahl eines Menüpunktes aus einem Menü z. B.: Datei > Neu
Eingabe	Bezeichnungen von Eingabe- oder Auswahlfeldern werden fett dargestellt z. B.: Messbereichsanfang
„Wert“	Eingabe- oder Auswahlwerte werden in Anführungszeichen dargestellt z. B.: Geben Sie unter Messbereichsanfang den Wert „4 mA“ ein.
[Button]	Schaltflächenbeschriftungen in Dialogen werden fett dargestellt und in eckigen Klammern eingefasst z. B.: [Eingabe]
[Taste]	Tastenbeschriftungen auf der Tastatur werden fett dargestellt und in eckigen Klammern eingefasst z. B.: [F5]

2 Wichtige Erläuterungen

Dieses Kapitel beinhaltet ausschließlich eine Zusammenfassung der wichtigsten Sicherheitsbestimmungen und Hinweise. Diese werden in den einzelnen Kapiteln wieder aufgenommen. Zum Schutz vor Personenschäden und zur Vorbeugung von Sachschäden an Geräten ist es notwendig, die Sicherheitsrichtlinien sorgfältig zu lesen und einzuhalten.

2.1 Rechtliche Grundlagen

2.1.1 Änderungsvorbehalt

Die WAGO Kontakttechnik GmbH & Co. KG behält sich Änderungen, die dem technischen Fortschritt dienen, vor. Alle Rechte für den Fall der Patenterteilung oder des Gebrauchsmusterschutzes sind der WAGO Kontakttechnik GmbH & Co. KG vorbehalten. Fremdprodukte werden stets ohne Vermerk auf Patentrechte genannt. Die Existenz solcher Rechte ist daher nicht auszuschließen.

2.1.2 Personalqualifikation

Sämtliche Arbeitsschritte, die an den Geräten der Serie 750 durchgeführt werden, dürfen nur von Elektrofachkräften mit ausreichenden Kenntnissen im Bereich der Automatisierungstechnik vorgenommen werden. Diese müssen mit den aktuellen Normen und Richtlinien für die Geräte und das Automatisierungsumfeld vertraut sein.

Alle Eingriffe in die Steuerung sind stets von Fachkräften mit ausreichenden Kenntnissen in der SPS-Programmierung durchzuführen.

2.1.3 Bestimmungsgemäße Verwendung der Serie 750

Feldbuskoppler, Feldbuscontroller und Busklemmen des modularen WAGO-I/O-SYSTEM 750 dienen dazu, digitale und analoge Signale von Sensoren aufzunehmen und an Aktoren auszugeben oder an übergeordnete Steuerungen weiterzuleiten. Mit den programmierbaren Feldbuscontrollern ist zudem eine (Vor-)Verarbeitung möglich.

Die Geräte sind für ein Arbeitsumfeld entwickelt, welches der Schutzklasse IP20 genügt. Es besteht Fingerschutz und Schutz gegen feste Fremdkörper bis 12,5 mm, jedoch kein Schutz gegen Wasser. Der Betrieb der Komponenten in nasser und staubiger Umgebung ist nicht gestattet, sofern nicht anders angegeben.

Not-Aus-Funktionen und -Einrichtungen dürfen nicht unwirksam gemacht werden. Beachten Sie dazu die einschlägigen Normen, z. B. die DIN EN 418.

Der Betrieb von Geräten der Serie 750 im Wohnbereich ist ohne weitere Maßnahmen nur zulässig, wenn diese die Emissionsgrenzen (Störaussendungen) gemäß EN 61000-6-3 einhalten. Entsprechende Angaben finden Sie im Kapitel

„Das WAGO-I/O-SYSTEM 750“ → „Systembeschreibung“ → „Technische Daten“ im Handbuch zum eingesetzten Feldbuskoppler/-controller.

Für den Betrieb des WAGO-I/O-SYSTEM 750 in explosionsgefährdeten Bereichen ist ein entsprechender Gehäuseschutz gemäß der Richtlinie 94/9/EG erforderlich. Zusätzlich ist zu beachten, dass eine Baumusterprüfbescheinigung erwirkt werden muss, die den korrekten Einbau des Systems im Gehäuse bzw. Schaltschrank bestätigt.

2.1.4 Technischer Zustand der Geräte

Die Geräte werden ab Werk für den jeweiligen Anwendungsfall mit einer festen Hard- und Software-Konfiguration ausgeliefert. Alle Veränderungen an der Hard- oder Software sowie der nicht bestimmungsgemäße Gebrauch der Komponenten bewirken den Haftungsausschluss der WAGO Kontakttechnik GmbH & Co. KG.

Wünsche an eine abgewandelte bzw. neue Hard- oder Software-Konfiguration richten Sie bitte an die WAGO Kontakttechnik GmbH & Co. KG.

2.2 Sicherheitshinweise

Beim Einbauen des Gerätes in Ihre Anlage und während des Betriebes sind folgende Sicherheitshinweise zu beachten:

GEFAHR**Nicht an Geräten unter Spannung arbeiten!**

Schalten Sie immer alle verwendeten Spannungsversorgungen für das Gerät ab, bevor Sie es montieren, Störungen beheben oder Wartungsarbeiten vornehmen.

GEFAHR**Einbau nur in Gehäusen, Schränken oder elektrischen Betriebsräumen!**

Das WAGO-I/O-SYSTEM 750 mit seinen Geräten ist ein offenes Betriebsmittel. Bauen Sie dieses ausschließlich in abschließbaren Gehäusen, Schränken oder in elektrischen Betriebsräumen auf. Ermöglichen Sie nur autorisiertem Fachpersonal den Zugang mittels Schlüssel oder Werkzeug.

GEFAHR**Unfallverhütungsvorschriften beachten!**

Beachten Sie bei der Montage, Inbetriebnahme, Wartung und Störbehebung die für Ihre Maschine zutreffenden Unfallverhütungsvorschriften wie beispielsweise die BGV A 3, „Elektrische Anlagen und Betriebsmittel“.

GEFAHR**Auf normgerechten Anschluss achten!**

Zur Vermeidung von Gefahren für das Personal und Störungen an Ihrer Anlage, verlegen Sie die Daten- und Versorgungsleitungen normgerecht und achten Sie auf die korrekte Anschlussbelegung. Beachten Sie die für Ihre Anwendung zutreffenden EMV-Richtlinien.

ACHTUNG**Nicht in Telekommunikationsnetzen einsetzen!**

Verwenden Sie Geräte mit ETHERNET-/RJ-45-Anschluss ausschließlich in LANs. Verbinden Sie diese Geräte niemals mit Telekommunikationsnetzen.

ACHTUNG**Defekte oder beschädigte Geräte austauschen!**

Tauschen Sie defekte oder beschädigte Geräte (z. B. bei deformierten Kontakten) aus, da die Funktion der betroffenen Feldbusstation langfristig nicht sichergestellt ist.

ACHTUNG**Geräte vor kriechenden und isolierenden Stoffen schützen!**

Die Geräte sind unbeständig gegen Stoffe, die kriechende und isolierende Eigenschaften besitzen, z. B. Aerosole, Silikone, Triglyceride (Bestandteil einiger Handcremes). Sollten Sie nicht ausschließen können, dass diese Stoffe im Umfeld der Geräte auftreten, bauen Sie die Geräte in ein Gehäuse ein, das resistent gegen oben genannte Stoffe ist. Verwenden Sie generell zur Handhabung der Geräte saubere Werkzeuge und Materialien.

ACHTUNG



Reinigung nur mit zulässigen Materialien!

Reinigen Sie verschmutzte Kontakte mit ölfreier Druckluft oder mit Spiritus und einem Ledertuch.

ACHTUNG



Kein Kontaktspray verwenden!

Verwenden Sie kein Kontaktspray, da in Verbindung mit Verunreinigungen die Funktion der Kontaktstelle beeinträchtigt werden kann.

ACHTUNG



Verpolungen vermeiden!

Vermeiden Sie die Verpolung der Daten- und Versorgungsleitungen, da dies zu Schäden an den Geräten führen kann.

ESD



Elektrostatische Entladung vermeiden!

In den Geräten sind elektronische Komponenten integriert, die Sie durch elektrostatische Entladung bei Berührung zerstören können. Beachten Sie die Sicherheitsmaßnahmen gegen elektrostatische Entladung gemäß DIN EN 61340-5-1/-3. Achten Sie beim Umgang mit den Geräten auf gute Erdung der Umgebung (Personen, Arbeitsplatz und Verpackung).

3 Programmierung des BACnet/IP-Controllers 750-830

Diese Referenz dient der Programmierung und Parametrierung von WAGO-BACnet/IP-Controllern 750-830 und beinhaltet folgende Punkte:

- Liste aller BACnet-Objekte für den nicht-nativen Betrieb
- Liste aller BACnet-Dienste
- Fehlercodes
- Alphabetische Liste aller Eigenschaften der BACnet-Objekte
- Alphabetische Liste aller Parameter der BACnet-Dienste

Information



Weitere Information

Die zu verwendende Bibliothek BACnet_xx.lib finden Sie auf der Internetseite <http://www.wago.com> unter Service → Downloads → Gebäudeautomation → BACnet Downloads → Software. Die zugehörige Dokumentation finden Sie auf der selben Seite.

Information



Weitere Information

Den WAGO-BACnet-Configurator und die zugehörige Dokumentation erhalten Sie auf der Internetseite <http://www.wago.com> unter Service → Downloads → Gebäudeautomation → BACnet Downloads → Software.

Hinweis



Maximale Objektanzahl beachten!

Im BACnet/IP-Controller 750-830 sind maximal 1000 Objekte anlegbar.

4 Objekte im nicht-nativen Betrieb

Der BACnet/IP-Controller unterstützt sämtliche WAGO-I/O-Klemmen. Analoge und digitale Ein-/Ausgangsklemmen sind direkt „nativ“ verfügbar, da die Objekte dieser Klemmen automatisch angelegt werden. Im nicht-nativen Betrieb legen Sie die BACnet-Objekte manuell in der IEC-Applikation an, programmieren und verknüpfen sie.

Nachfolgend erhalten Sie eine Beschreibung der Objekte, die im nicht-nativen Betrieb angelegt werden können. Jedes Objekt beinhaltet Eigenschaften, die tabellarisch dargestellt und in Kapitel „Gesamtliste der BACnet-Objekt-Eigenschaften“ detailliert beschrieben werden.

4.1 Accumulator-Objekt

Das Accumulator-Objekt stellt von Außen erkennbare Merkmale eines Gerätes dar, das Messungen durch das Zählen von Impulsen angibt.

Dieses Objekt garantiert eine genaue Messung der Eingangszählerwerte über den Zeitverlauf. Die Anzahl von Impulsen stellt die gemessene Menge als Unsigned-Integer-Wert dar. Das Accumulator-Objekt ist auch für die genaue Darstellung von Werten auf Messgeräten zuständig. Dadurch kann die Eigenschaft `Present_Value` so an den vom Messgerät angezeigten Wert angepasst werden, als ob das Gerät angeschlossen wäre. Die Methode der Zählung, die Simulation eines Modulo-N-Teilers und Anpassung der Messgeräteanzeige wird imitiert (siehe folgende Abbildung).

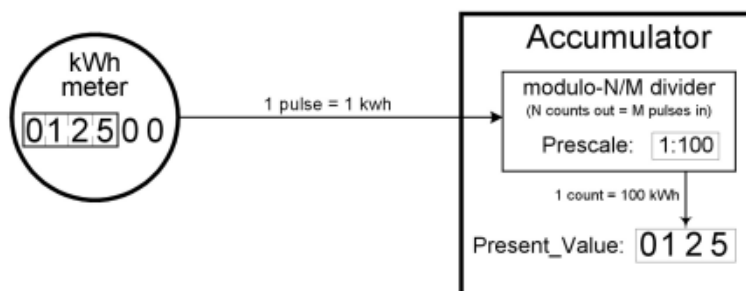


Abbildung 1: Beispiel eines Accumulator-Objektes

Das Objekt und dessen Eigenschaften werden in der folgenden Tabelle zusammengefasst.

Tabelle 3: Eigenschaften des Accumulator-Objektes

Eigenschaften	Datentyp	Basiert auf IEC-Datentypen	Vorgabewert	Schreibbar via BACnet mittels
Object_Identifier	BACnetObjectIdentifier	BACnetObjectIdentifier	-	-
Object_Name	CharacterString	STRING	“AccumulatorX”	-
Object_Type	BACnetObjectType	BACnetObjectType	ACCUMULATOR (23)	-
Present_Value	Unsigned	WORD	-	-
Description	CharacterString	STRING	“”	WriteProperty
Device_Type	CharacterString	STRING	“”	WriteProperty
Status_Flags	BACnetStatusFlags	BACnetStatusFlags	‘0000’	-
Event_State	BACnetEventState	BACnetEventState	-	-
Reliability	BACnetReliability	BACnetReliability	-	WriteProperty
Out_Of_Service	BOOLEAN	BOOL	-	WriteProperty
Scale	BACnetScale	BACnetScale	1.0	WriteProperty
Units	BACnetEngineeringUnits	BACnetEngineeringUnits	None (95)	WriteProperty
Prescale	BACnetPrescale	BACnetPrescale	-	WriteProperty
Max_Pres_Value	Unsigned	WORD	-	-
Value_Change_Time	BACnetDateTime	BACnetDateTime	-	-
Value_Before_Change	Unsigned	WORD	-	-
Value_Set	Unsigned	WORD	-	WriteProperty
Pulse_Rate	Unsigned	WORD	-	-
High_Limit	Unsigned	WORD	-	WriteProperty
Low_Limit	Unsigned	WORD	-	WriteProperty
Limit_Monitoring_Interval	Unsigned	WORD	60	WriteProperty
Notification_Class	Unsigned	WORD	-	WriteProperty
Time_Delay	Unsigned	WORD	-	WriteProperty
Limit_Enable	BACnetLimitEnable	BACnetLimitEnable	‘11’	WriteProperty
Event_Enable	BACnetEventTransitionBits	BACnetEventTransitionBits	‘111’	WriteProperty
Acked_Transitions	BACnetEventTransitionBits	BACnetEventTransitionBits	‘111’	-
Notify_Type	BACnetNotifyType	BACnetNotifyType	-	WriteProperty
Event Time Stamps	BACnetARRAY[3] of BACnetTimeStamp	ARRAY [1..3] of BACnetTimeStamp	UNSPECIFIED	-

4.2 Analog-Value-Objekt

Ein Analog-Value ist ein Steuerungsparameter im Speicher des BACnet-Gerätes. Das Objekt und dessen Eigenschaften werden in der folgenden Tabelle zusammengefasst.

Tabelle 4: Eigenschaften des Analog-Value-Objektes

Eigenschaften	Datentyp	Basiert auf IEC-Datentypen	Vorgabewert	Schreibbar via BACnet mittels
Object_Identifier	BACnetObjectIdentifier	BACnetObjectIdentifier	-	-
Object_Name	CharacterString	STRING	“AnalogValueX”	-
Object_Type	BACnetObjectType	BACnetObjectType	ANALOG_VALUE (2)	-
Present_Value	REAL	REAL	-	WriteProperty
Description	CharacterString	STRING	“”	WriteProperty
Status_Flags	BACnetStatusFlags	BACnetStatusFlags	‘0000’	-
Event_State	BACnetEventState	BACnetEventState	-	-
Reliability	BACnetReliability	BACnetReliability	-	-
Out_Of_Service	BOOLEAN	BOOL	-	-
Units	BACnetEngineeringUnits	BACnetEngineeringUnits	None (95)	WriteProperty
Priority_Array	BACnetPriorityArray	ARRAY [1..16] OF BOOL	-	-
Relinquish_Default	REAL	REAL	-	-
COV_Increment	REAL	REAL	-	WriteProperty
Time_Delay	Unsigned	WORD	-	WriteProperty
Notification_Class	Unsigned	WORD	-	WriteProperty
High_Limit	REAL	REAL	-	WriteProperty
Low_Limit	REAL	REAL	-	WriteProperty
Deadband	REAL	REAL	-	WriteProperty
Limit_Enable	BACnetLimitEnable	BACnetLimitEnable	‘11’	WriteProperty
Event_Enable	BACnetEventTransitionBits	BACnetEventTransitionBits	-	WriteProperty
Acked_Transitions	BACnetEventTransitionBits	BACnetEventTransitionBits	‘111’	-
Notify_Type	BACnetNotifyType	BACnetNotifyType	-	WriteProperty
Event Time Stamps	BACnetARRAY[3] of BACnetTimeStamp	ARRAY [1..3] OF BACnetTimeStamp	UNSPECIFIED	—

4.3 Averaging-Objekt

Das Averaging-Objekt stellt einen Wert dar, der regelmäßig in einem bestimmten Intervall abgefragt wird. Ein Averaging-Objekt speichert Minimal-, Maximal- und Durchschnittswert des Intervalls und zeigt diese Werte als Eigenschaft des Averaging-Objektes an. Der abgefragte Wert kann eine BOOLEAN-, INTEGER-, Unsigned-, Enumerated- oder REAL-Eigenschaft eines beliebigen Objektes des BACnet-Gerätes sein, in dem sich das Objekt befindet. Optional kann sich die abgefragte Objekteigenschaft auch in einem anderen BACnet-Gerät befinden. Das Averaging-Objekt verwendet eine Art „Schieberegister“ und speichert N Werte pro Intervall. Alle (Intervall/N) Sekunden wird ein neues Sample abgefragt und ersetzt das älteste Sample im Speicher. Zu diesem Zeitpunkt werden auch Minimal-, Maximal- und Durchschnittswert neu berechnet. Der Speicher enthält Angaben zu jedem Sample, die eine Berechnung des Durchschnittswertes sowie die Ausführung eines Minimum/Maximum-Algorithmus zur Bestimmung der Anzahl gültiger Samples im Speicher ermöglichen.

Das Averaging-Objekt und dessen Eigenschaften werden in der folgenden Tabelle zusammengefasst.

Tabelle 5: Eigenschaften des Averaging-Objektes

Eigenschaften	Datentyp	Basiert auf IEC-Datentypen	Vorgabewert	Schreibbar via BACnet mittels
Object_Identifier	BACnetObjectIdentifier	BACnetObjectIdentifier	-	-
Object_Name	CharacterString	STRING	“AveragingX”	-
Object_Type	BACnetObjectType	BACnetObjectType	AVERAGING	-
Minimum_Value	REAL	REAL	-	-
Minimum_Value_Timestamp	BACnetDateTime	BACnetDateTime	-	-
Average_Value	REAL	REAL	-	-
Maximum_Value	REAL	REAL	-	-
Maximum_Value_Timestamp	BACnetDateTime	BACnetDateTime	-	-
Description	CharacterString	STRING	“”	WriteProperty
Attempted_Samples	Unsigned	WORD	-	WriteProperty
Valid_Samples	Unsigned	WORD	-	-
Object_Property_Reference	BACnetDeviceObject-PropertyReference	BACnetDeviceObject-PropertyReference	-	WriteProperty
Window_Interval	Unsigned	WORD	-	WriteProperty
Window_Samples	Unsigned	WORD	-	WriteProperty

4.4 Binary-Value-Objekt

Ein Binary-Value ist ein Controller-Parameter im Speicher des BACnet-Gerätes. Dieser Parameter kann nur einen von zwei eindeutigen Zuständen annehmen. In dieser Beschreibung werden diese Zustände mit ACTIVE oder INACTIVE bezeichnet. Das Binary-Value-Objekt und dessen Eigenschaften werden in der folgenden Tabelle zusammengefasst.

Tabelle 6: Eigenschaften des Binary-Value-Objektes

Eigenschaften	Datentyp	Basiert auf IEC-Datentypen	Vorgabewert	Schreibbar via BACnet mittels
Object_Identifier	BACnetObjectIdentifier	BACnetObjectIdentifier	-	-
Object_Name	CharacterString	STRING	“BinaryValueX”	-
Object_Type	BACnetObjectType	BACnetObjectType	BINARY_VALUE	-
Present_Value	BACnetBinaryPV	BACnetBinaryPV	-	WriteProperty
Description	CharacterString	STRING	“”	WriteProperty
Status_Flags	BACnetStatusFlags	BACnetStatusFlags	‘0000’	-
Event_State	BACnetEventState	BACnetEventState	-	-
Reliability	BACnetReliability	BACnetReliability	-	WriteProperty
Out_Of_Service	BOOLEAN	BOOL	-	WriteProperty
Inactive_Text	CharacterString	STRING	“inactive”	WriteProperty
Active_Text	CharacterString	STRING	“active”	WriteProperty
Change_Of_State_Time	BACnetDateTime	BACnetDateTime	UNSPECIFIED	-
Change_Of_State_Count	Unsigned32	DWORD	-	WriteProperty
Time_Of_State_Count_Reset	BACnetDateTime	BACnetDateTime	UNSPECIFIED	-
Elapsed_Active_Time	Unsigned32	DWORD	-	WriteProperty
Time_Of_Active_Time_Reset	BACnetDateTime	BACnetDateTime	UNSPECIFIED	-
Minimum_Off_Time	Unsigned32	DWORD	1	WriteProperty
Minimum_On_Time	Unsigned32	DWORD	1	WriteProperty
Priority_Array	BACnetPriorityArray	ARRAY [1..16] OF BOOL;	-	-
Relinquish_Default	BACnetBinaryPV	BACnetBinaryPV	-	WriteProperty
Time_Delay	Unsigned	WORD	-	WriteProperty
Notification_Class	Unsigned	WORD	-	WriteProperty
Alarm_Value	BACnetBinaryPV	BACnetBinaryPV	1	WriteProperty
Event_Enable	BACnetEventTransition-Bits	BACnetEventTransitionBits	‘111’	WriteProperty
Acked_Transitions	BACnetEventTransition-Bits	BACnetEventTransitionBits	‘111’	-
Notify_Type	BACnetNotifyType	BACnetNotifyType	-	WriteProperty
Event Time Stamps	BACnetARRAY[3] of BACnetTimeStamp	ARRAY [1..3] OF BACnetTimeStamp	UNSPECIFIED	-

4.5 Calendar-Objekt

Das Calendar-Objekt definiert eine Liste bestimmter Daten, wie Urlaubszeiten, Feiertagen und anderen besonderen Ereignissen. Das Objekt und dessen Eigenschaften werden in der folgenden Tabelle zusammengefasst.

Tabelle 7: Eigenschaften des Calendar-Objektes

Eigenschaften	Datentyp	Basiert auf IEC-Datentypen	Vorgabewert	Schreibbar via BACnet mittels
Object_Identifier	BACnetObjectIdentifier	BACnetObjectIdentifier	-	-
Object_Name	CharacterString	STRING	“CalendarX”	-
Object_Type	BACnetObjectType	BACnetObjectType	CALENDAR	-
Present_Value	BOOLEAN	BOOL	-	WriteProperty
Description	CharacterString	STRING	“”	WriteProperty
Date_List	List of BACnetCalendarEntry	ARRAY[0..3] OF BACnetCalendarEntry;	-	WriteProperty

4.6 Command-Objekt

Das Command-Objekt definiert einen Vorgang, welcher mehrere Schreib-Befehle beim Schreiben in die Eigenschaft `Present_Value` auslöst.

Ein Command-Objekt wird in der Regel verwendet, um einen komplexen Kontext mit mehreren Variablen darzustellen. Command-Objekte sind besonders hilfreich für die Darstellung von Kontexten mit mehreren Zuständen. Ein bestimmter Bereich eines Gebäudes kann beispielsweise die drei Zustände UNOCCUPIED, WARMUP und OCCUPIED annehmen. Um den Betriebskontext für jeden Zustand festzulegen, ist unter Umständen die Änderung einer Vielzahl von Objekteigenschaften auf bestimmte Werte erforderlich. Im Zustand „Unoccupied“ ist die Temperatureinstellung beispielsweise 18 °C und die Beleuchtung ist ausgeschaltet, im Zustand „Occupied“ beträgt die Einstellung 22 °C und die Beleuchtung ist eingeschaltet usw.

Das Command-Objekt legt das Verhältnis zwischen einem bestimmten Zustand und den zugehörigen Werten fest, die für diesen Zustand in eine Gruppe verschiedener Objekteigenschaften geschrieben werden. In der Regel ist das Command-Objekt passiv. Die Eigenschaft `In_Process` ist auf „FALSE“ gesetzt und gibt an, dass das Command-Objekt darauf wartet, dass ein Wert in die Eigenschaft `Present_Value` geschrieben wird. Ist `Present_Value` geschrieben, beginnt das Command-Objekt mit der Ausführung der definierten Aktionen. Die Eigenschaft `In_Process` wird auf „TRUE“ gesetzt und gibt an, dass das Command-Objekt mit der Bearbeitung einer Aktionssequenz begonnen hat, die durch den in die Eigenschaft `Present_Value` geschriebenen Wert ausgewählt wurde. Bei dem Versuch, mit dem Dienst `WriteProperty` in die Eigenschaft `Present_Value` zu schreiben, während `In_Process` „TRUE“ ist, wird ein Fehler zurückgegeben und das Schreiben verhindert.

Der neue Wert der Eigenschaft `Present_Value` bestimmt, welche Aktionssequenz das Command-Objekt ausführt. Die Aktionen sind in einem Feld mit Aktionslisten gespeichert, die dieser Wert indiziert. Die Listen sind in der Eigenschaft `Action` enthalten. Eine Liste kann auch leer sein, in diesem Fall wird keine Aktion ausgeführt, außer dass `In_Process` zurück auf „FALSE“ und `All_Writes_Successful` auf „TRUE“ gesetzt werden. Ist die Liste nicht leer, schreibt das Command-Objekt für jede Aktion in der Liste einen bestimmten Wert in eine bestimmte Eigenschaft eines bestimmten Objektes eines bestimmten BACnet-Gerätes.

Das Command-Objekt garantiert allerdings nicht, dass jeder Schreibvorgang erfolgreich ist, und es versucht nicht, erfolgreich geschriebene Eigenschaften auf ihre ursprünglichen Werte zurückzusetzen, falls einer oder mehrere Schreibvorgänge fehlschlagen. Schlägt einer oder mehrere Schreibvorgänge fehl, werden die Eigenschaft `All_Writes_Successful` und das Flag `Write_Successful` für diesen `BACnetActionCommand` auf „FALSE“ gesetzt. Ist das `Quit_On_Failure` Flag für den fehlgeschlagenen

BACnetActionCommand „TRUE“, werden die Write_Successful Flags aller weiteren BACnetActionCommands auf „FALSE“ gesetzt. Ist ein einzelner Schreibvorgang erfolgreich, wird das Write_Successful Flag für diesen BACnetActionCommand auf „TRUE“ gesetzt. Sind alle Schreibvorgänge erfolgreich, wird die Eigenschaft All_Writes_Successful auf „TRUE“ gesetzt. Nach Abschluss der Ausführung aller Schreibvorgänge des Command-Objektes wird die Eigenschaft In_Process zurück auf „FALSE“ gesetzt und das Command-Objekt ist wieder passiv und wartet auf einen neuen Befehl.

Es ist zu beachten, dass nicht der in die Eigenschaft Present_Value geschriebene Wert die Aktion auslöst, sondern der Schreibvorgang selbst. Hat die Eigenschaft Present_Value beispielsweise den Wert 5 und wird erneut mit dem Wert 5 geschrieben, wird die fünfte Aktionsliste erneut ausgeführt. Beim Schreiben von Null in Present_Value wird keine Aktion ausgeführt, genauso wie bei einer leeren Aktionsliste.

Das Command-Objekt ist ein leistungsfähiges Konzept mit vielen nützlichen Anwendungsmöglichkeiten. Es gibt jedoch einzelne Aspekte des Command-Objektes, die verwirrende oder schädliche Auswirkungen haben können, falls das Command-Objekt nicht richtig konfiguriert ist. Da das Command-Objekt die Eigenschaften anderer Objekte verändern kann, ist es möglich, dass das Command-Objekt so konfiguriert ist, dass es sich selbst verändert. In diesem Fall fungiert die Eigenschaft In_Process als Sperre und schützt das Command-Objekt vor einer Bearbeitung. Ein Command-Objekt kann jedoch ein anderes Command-Objekt verändern, das wiederum das erste Command-Objekt verändert usw. Command-Objekte können Group-Objekte verändern. Derartige Fälle von „Zirkelverweisen“ können zu verwirrenden Nebenwirkungen führen. Bei Verweisen auf Objekte in anderen BACnet-Geräten besteht die Gefahr von Verzögerungen, die zu instabilem Verhalten von Command-Objekten führen können, die in solchen Zirkelverweisen nicht richtig konfiguriert sind. Die Konfiguration von Command-Objekten, die auf Objekte außerhalb des enthaltenden BACnet-Gerätes verweisen, ist deshalb äußerst sorgfältig vorzunehmen. Das Command-Objekt und dessen Eigenschaften werden in der folgenden Tabelle zusammengefasst.

Tabelle 8: Eigenschaften des Command-Objektes

Eigenschaften	Datentyp	Basiert auf IEC-Datentypen	Vorgabewert	Schreibbar via BACnet mittels
Object_Identifier	BACnetObjectIdentifier	BACnetObjectIdentifier	-	-
Object_Name	CharacterString	STRING	„CommandX“	-
Object_Type	BACnetObjectType	BACnetObjectType	COMMAND	-
Present_Value	Unsigned	WORD	-	WriteProperty
Description	CharacterString	STRING	“”	WriteProperty
In_Process	BOOLEAN	BOOL	-	-
All_Writes_Successful	BOOLEAN	BOOL	TRUE	-
Action	BACnetARRAY[N] of BACnetActionList	ARRAY [1..3] OF BACnetActionList;	-	WriteProperty
Action_Text	BACnetARRAY[N] of CharacterString	ARRAY [1..3] OF STRING;	-	WriteProperty

4.7 File-Objekt

Das File-Objekt repräsentiert Dateien, auf die mithilfe der File-Dienste zugegriffen werden kann. Das File-Objekt und dessen Eigenschaften werden in der folgenden Tabelle zusammengefasst.

Tabelle 9: Eigenschaften des File-Objektes

Eigenschaften	Datentyp	Basiert auf IEC-Datentypen	Vorgabewert	Schreibbar via BACnet mittels
Object_Identifier	BACnetObjectIdentifier	BACnetObjectIdentifier	-	-
Object_Name	CharacterString	STRING	“FileX”	-
Object_Type	BACnetObjectType	BACnetObjectType	FILE	-
Description	CharacterString	STRING	“”	WriteProperty
File_Type	CharacterString	STRING	“”	WriteProperty
File_Size	Unsigned	WORD	-	WriteProperty
Modification_Date	BACnetDateTime	BACnetDateTime	UNSPECIFIED	-
Archive	BOOLEAN	BOOL	-	WriteProperty
Read_Only	BOOLEAN	BOOL	TRUE	WriteProperty
File_Access_Method	BACnetFileAccessMethod	BACnetFileAccessMethod	STREAM_ACCESS	-

4.8 Group-Objekt

Das Group-Objekt entspricht einer Sammlung von Eigenschaften aus verschiedenen Objekten. Ein Group-Objekt wird verwendet, um den Informationsaustausch zwischen BACnet-Geräten zu vereinfachen, indem eine Möglichkeit für die gleichzeitige Angabe aller Gruppenmitglieder geschaffen wird. Eine Gruppe kann aus einer beliebigen Kombination von Objekten bestehen. Das Group-Objekt und dessen Eigenschaften werden in der folgenden Tabelle zusammengefasst.

Tabelle 10: Eigenschaften des Group-Objektes

Eigenschaften	Datentyp	Basiert auf IEC-Datentypen	Vorgabewert	Schreibbar via BACnet mittels
Object_Identifier	BACnetObjectIdentifier	BACnetObjectIdentifier	-	-
Object_Name	CharacterString	STRING	“GroupX”	-
Object_Type	BACnetObjectType	BACnetObjectType	“GroupX”	-
Description	CharacterString	STRING	“”	WriteProperty
List_Of_Group_Members	List of ReadAccessSpecification	ARRAY [0..3] OF ReadAccessSpecification;	.	WriteProperty
Present_Value	List of ReadAccessResult	ARRAY [0..3] OF ReadAccessResult;	-	-

4.9 Life-Safety-Point-Objekt

Das Life-Safety-Point-Objekt repräsentiert die Merkmale, welche für die Initiierung und Anzeige von Geräten in Brand-, Personenschutz- und Sicherheitsanwendungen verwendet werden. Der Zustand eines Life-Safety-Point-Objektes wird durch eine Betriebsart (*Mode*) und einen Zustand (*State*) dargestellt.

Änderungen der Betriebsart bestimmen die interne Logik des Objektes und beeinflussen so die Beurteilung des Zustandes. In der Regel unterliegt die Betriebsart der Kontrolle des Betreibers.

Die Kerneigenschaften der Life-Safety-Point-Objekte sind:

- `Present_Value` (der eventuell verlinkte Status des Life-Safety-Point)
- `Tracking_Value` (der unverlinkte aktuelle Status des Life-Safety-Point)
- `Mode` (die Betriebsart des Life-Safety-Point)
- `Silenced` (der Status einer sichtbaren oder hörbaren Anzeige, die mit dem Life-Safety-Point verbunden ist)
- `Operation_Expected` (die spezielle Anfrage, die über den Life-SafetyOperation-Dienst erwartet wird, in Abhängigkeit von den Werten aus `Present_Value`, `Modus` und `Silenced`)

Da die Bedeutung der Eigenschaften von `Present_Value`, `Tracking_Value`, `Modus` und `Silenced` eng mit dem speziellen Baustein oder Prozess verbunden ist, der durch das Life-Safety-Point-Objekt dargestellt wird, können diese Werte nicht innerhalb des C/S-Moduls gespeichert werden.

`Operation_Expected` ist normalerweise eine Funktion von `Present_Value`, `Mode` und `Silenced`. Der Wert von `Operation_Expected` wird vorzugsweise durch den Baustein Life-Safety-Point selbst festgelegt. Sollte letzteres nicht der Fall sein, wird der Wert von `Operation_Expected` gemäß folgender Tabelle festgelegt.

Tabelle 11: Present Value des Life-Safety-Point-Objektes

Present Value	Mode	Silenced	Operation Expected
*	off disabled	all-silenced	none
*	off disabled	unsilenced audible-silenced visible-silenced	silence
quiet active supervisory test-active test-supervisory emergency-power	!off & !disabled	all-silenced	none
quiet active supervisory test-active test-supervisory emergency-power	!off & !disabled	unsilenced audible-silenced visible-silenced	silence
alarm pre-alarm local-alarm general-alarm tamper-alarm tamper	!off & !disabled	all-silenced audible-silenced visible-silenced	unsilence
alarm pre-alarm local-alarm general-alarm tamper-alarm tamper	!off & !disabled	unsilenced	reset-alarm
fault fault-pre-alarm fault-alarm test-fault test-fault-alarm	!off & !disabled	*	reset-fault
not-ready holdup duress abnormal delayed blocked	!off & !disabled	*	reset
Keiner der obigen Bedingungen ist erfüllt. Dies passiert nur, wenn Present_Value oder Silenced einen Wert annehmen, der nicht als Name im Standard festgelegt ist.			none

Der Status eines Life-Safety-Point-Objektes, d. h. der Present_Value sowie jegliche sichtbare oder hörbare Anzeige, die mit dem Life-Safety-Point-Objekt verbunden ist (Silenced), können über die Life-Safety-Operation-Dienstanfragen geändert werden. Diese werden von den entfernten BACnet-Client-Geräten übertragen.

Typische Anwendungen des Life-Safety-Point-Objektes sind beispielsweise automatische Brandmelder, manuelle Feueralarme, Sirenen, überwachte Drucker usw. Ähnliche Objekte werden in Sicherheitsschalttafeln verwendet.

Das Life-Safety-Point-Objekt und dessen Eigenschaften werden in der folgenden Tabelle zusammengefasst.

ACHTUNG



BACnet/IP-Controller nicht für sicherheitsrelevante Anwendungen verwenden!

Der Controller erfüllt nicht die Anforderungen an die IEC 61508-Norm und ist nicht für sicherheitskritische Anwendungen zu verwenden.

Hinweis



Zustand des Present_Value nicht mit Event_State verwechseln!

Verwechseln Sie den Zustand des Present_Value nicht mit der Eigenschaft Event_State, die den OFFNORMAL-Zustand des Life-Safety-Point-Objektes angibt.

Tabelle 12: Eigenschaften des Life-Safety-Point-Objektes

Eigenschaften	Datentyp	Basiert auf IEC-Datentypen	Vorgabewert	Schreibbar via BACnet mittels
Object_Identifier	BACnetObjectIdentifier	BACnetObjectIdentifier	-	-
Object_Name	CharacterString	STRING	“LifeSafety-PointX”	-
Object_Type	BACnetObjectType	BACnetObjectType	LIFE_SAFETY_POINT	-
Present_Value	BACnetLifeSafetyState	BACnetLifeSafetyState	-	-
Description	CharacterString	STRING	“”	WriteProperty
Tracking_Value	BACnetLifeSafetyState	BACnetLifeSafetyState	-	-
Device_Type	CharacterString	STRING	“”	WriteProperty
Status_Flags	BACnetStatusFlags	BACnetStatusFlags	‘0000’	-
Event_State	BACnetEventState	BACnetEventState	-	-
Reliability	BACnetReliability	BACnetReliability	-	WriteProperty
Out_Of_Service	BOOLEAN	BOOL	-	WriteProperty
Mode	BACnetLifeSafetyMode	BACnetLifeSafetyMode	-	WriteProperty
Accepted_Modes	List of BACnetLifeSafetyMode	ARRAY [0..3] OF BACnetLifeSafetyMode;	-	WriteProperty
Time_Delay	Unsigned	WORD	-	WriteProperty
Notification_Class	Unsigned	WORD	-	WriteProperty
Life_Safety_Alarm_Values	List of BACnetLifeSafetyState	ARRAY [0..3] OF BACnetLifeSafetyState;	-	WriteProperty
Alarm_Values	List of BACnetLifeSafetyState	ARRAY [0..3] OF BACnetLifeSafetyState;	-	WriteProperty
Fault_Values	List of BACnetLifeSafetyState	ARRAY [0..3] OF BACnetLifeSafetyState;	-	WriteProperty
Event_Enable	BACnetEventTransitionBits	BACnetEventTransitionBits	-	WriteProperty
Acked_Transitions	BACnetEventTransitionBits	BACnetEventTransitionBits	‘111’	-
Notify_Type	BACnetNotifyType	BACnetNotifyType	-	WriteProperty
Event Time Stamps	BACnetARRAY [3] of BACnetTimeStamp	ARRAY [1..3] OF BACnetTimeStamp;	UNSPECIFIED	-
Silenced	BACnetSilencedState	BACnetSilencedState	-	-
Operation_Expected	BACnetLifeSafetyOperation	BACnetLifeSafetyOperation	-	-
Maintenance_Required	BACnetMaintenance	BACnetMaintenance	-	-
Setting	Unsigned8	BYTE	-	WriteProperty
Direct_Reading	REAL	REAL	-	WriteProperty
Units	BACnetEngineeringUnits	BACnetEngineeringUnits	None (95)	WriteProperty
Member_Of	List of BACnetDevice-ObjectReference	ARRAY [0..3] OF BACnetDeviceObjectReference;	-	WriteProperty

4.10 Life-Safety-Zone-Objekt

Eine Life-Safety-Zone ist eine Gruppierung von Life-Safety-Points und Life-Safety-Zones zu einer logischen Einheit. Diese logische Einheit wird durch das Life-Safety-Zone-Objekt konfiguriert. Der Zustand eines Life-Safety-Zone-Objektes wird wie das Life-Safety-Point-Objekt durch eine Betriebsart (Mode) und einen Zustand (State) dargestellt.

Typische Anwendungen des Life-Safety-Zone-Objektes sind beispielsweise Brandbereiche, Schalttafelbereiche, Brandmeldeleitungen, Löschcontroller, entfernte Controller für Übertragungen usw. Ähnliche Objekte werden in Sicherheitsschalttafeln verwendet.

Das Life-Safety-Zone-Objekt und dessen Eigenschaften werden in der folgenden Tabelle zusammengefasst.

Das Hasse-Diagramm der vorgegebenen Halbordnung

Für zwei direkt angeschlossene Knoten ist der eine Knoten, der höher in dem Baum liegt, der größere. Beispiel: Alarm ³ lokaler Alarm. Der Status “*” zeigt alle Zustände an, die nicht durch den Standard spezifiziert werden. Bitte beachten Sie, dass nicht alle Zustände vergleichbar sein müssen, siehe folgende Abbildung.

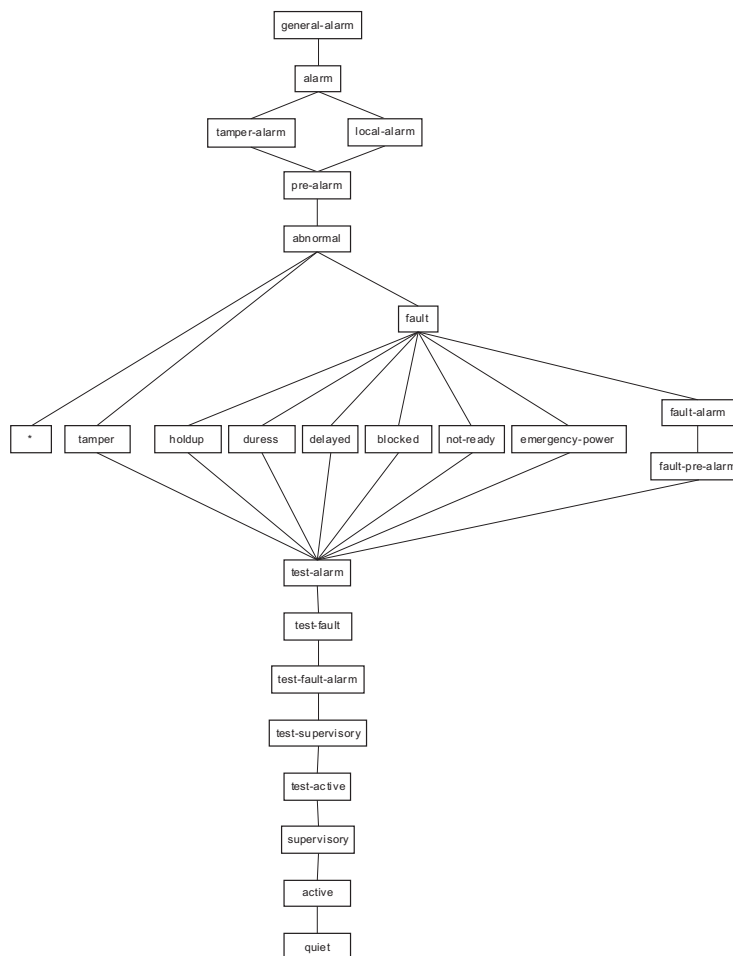


Abbildung 2: Hasse-Diagramm

Die Hasse-Diagramme der vorgegebenen Halbordnungen für die Eigenschaften „Silenced“ und „Operation_Expected“ werden in den folgenden zwei Abbildungen dargestellt.

Hasse-Diagramm der vorgegebenen Halbordnung, die zur Ableitung des Wertes von “Silenced” für einen Life Safety-Bereich genutzt wird

“*” kennzeichnet jeden beliebigen Wert für “Silenced”, der in dem Standard nicht genannt wird (siehe folgende Abbildung).

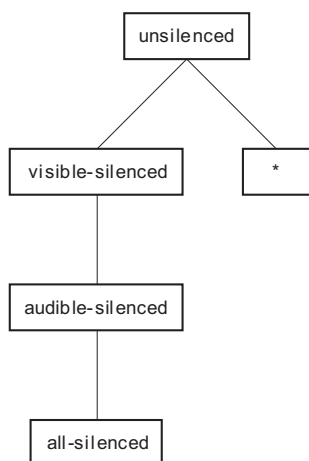


Abbildung 3: Hasse-Diagramm (Silenced)

Hasse-Diagramm der vorgegebenen Halbordnung, die zur Ableitung des Wertes von Operation_Expected für einen Life-Safety-Bereich genutzt wird

“*” kennzeichnet jeden beliebigen Wert für Operation_Expected, der in dem Standard nicht genannt wird (siehe folgende Abbildung).

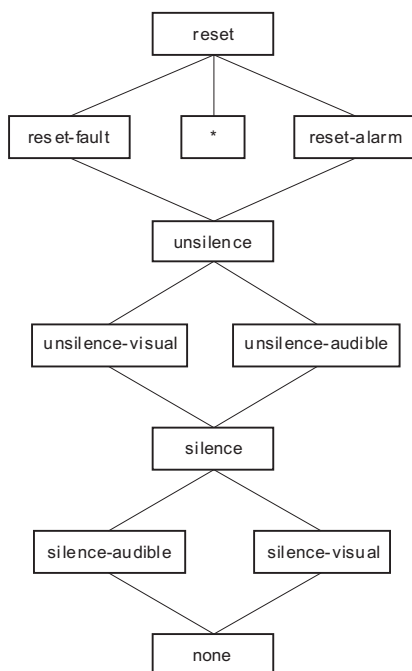


Abbildung 4: Hasse-Diagramm (Operation_Expected)

ACHTUNG**Nicht für sicherheitskritische Anwendungen verwenden!**

Der BACnet/IP-Controller erfüllt nicht die Anforderungen an die IEC 61508-Norm und ist nicht für sicherheitskritische Anwendungen zu verwenden.

Hinweis**Zustand des Present_Value nicht mit Event_State verwechseln!**

Verwechseln Sie den Zustand des Present_Value nicht mit der Eigenschaft Event_State, die den OFFNORMAL-Zustand des Life-Safety-Zone-Objektes angibt.

Tabelle 13: Eigenschaften des Life-Safety-Zone-Objektes

Eigenschaften	Datentyp	Basiert auf IEC-Datentypen	Vorgabewert	Schreibbar via BACnet mittels
Object_Identifier	BACnetObjectIdentifier	BACnetObjectIdentifier	-	-
Object_Name	CharacterString	STRING	“LifeSafety-ZoneX”	-
Object_Type	BACnetObjectType	BACnetObjectType	LIFE_SAFETY_ZONE	-
Present_Value	BACnetLifeSafetyState	BACnetLifeSafetyState	-	-
Tracking_Value	BACnetLifeSafetyState	BACnetLifeSafetyState	-	-
Description	CharacterString	STRING	“”	WriteProperty
Device_Type	CharacterString	STRING	“”	WriteProperty
Status_Flags	BACnetStatusFlags	BACnetStatusFlags	‘0000’	-
Event_State	BACnetEventState	BACnetEventState	-	-
Reliability	BACnetReliability	BACnetReliability	-	WriteProperty
Out_Of_Service	BOOLEAN	BOOL	-	WriteProperty
Mode	BACnetLifeSafetyMode	BACnetLifeSafetyMode	-	WriteProperty
Accepted_Modes	List of BACnetLifeSafetyMode	ARRAY [0..3] OF BACnetLifeSafetyMode;	-	WriteProperty
Time_Delay	Unsigned	WORD	-	WriteProperty
Notification_Class	Unsigned	WORD	-	WriteProperty
Life_Safety_Alarm_Values	List of BACnetLifeSafetyState	ARRAY [0..3] OF BACnetLifeSafetyState;	-	WriteProperty
Alarm_Values	List of BACnetLifeSafetyState	ARRAY [0..3] OF BACnetLifeSafetyState;	-	WriteProperty
Fault_Values	List of BACnetLifeSafetyState	ARRAY [0..3] OF BACnetLifeSafetyState;	-	WriteProperty
Event_Enable	BACnetEventTransitionBits	BACnetEventTransitionBits	‘111’	WriteProperty
Acked_Transitions	BACnetEventTransitionBits	BACnetEventTransitionBits	‘111’	-
Notify_Type	BACnetNotifyType	BACnetNotifyType		WriteProperty
Event Time Stamps	BACnetARRAY [3] of BACnetTimeStamp	ARRAY [1..3] OF BACnetTimeStamp;	UNSPECIFIED	-
Silenced	BACnetSilencedState	BACnetSilencedState	-	-
Operation_Expected	BACnetLifeSafetyOperation	BACnetLifeSafetyOperation	-	-
Maintenance_Required	BOOLEAN	BOOL	-	WriteProperty
Zone_Members	List of BACnetDevice-ObjectReference	ARRAY [0..3] OF BACnetDeviceObjectReference;	-	WriteProperty
Member_Of	List of BACnetDevice-ObjectReference	ARRAY [0..3] OF BACnetDeviceObjectReference;	-	WriteProperty

4.11 Multi-State-Value-Objekt

Ein Multi-State-Value ist ein Controller-Parameter im Speicher des BACnet-Gerätes. Die tatsächlichen Funktionen eines bestimmten Zustandes sind nicht definiert. Ein Zustand kann beispielsweise Aktivität oder Inaktivität mehrerer physikalischer Ein- und Ausgänge oder den Wert eines analogen Ein- oder Ausganges angeben. Die Eigenschaft `Present_Value` gibt den Unsigned-Integer-Wert des Zustandes an. Die Eigenschaft `State_Text` liefert eine Beschreibung für jeden Zustand.

Das Multi-State-Value-Objekt und dessen Eigenschaften werden in der folgenden Tabelle zusammengefasst.

Hinweis



Zustand des `Present_Value` nicht mit `Event_State` verwechseln!

Verwechseln Sie den Zustand von `Present_Value` nicht mit der Eigenschaft `Event_State`, die den OFFNORMAL-Zustand des Multi-State-Value-Objektes angibt.

Tabelle 14: Eigenschaften des Multi-State-Value-Objektes

Eigenschaften	Datentyp	Basiert auf IEC-Datentypen	Vorgabewert	Schreibbar via BACnet mittels
Object_Identifier	BACnetObjectIdentifier	BACnetObjectIdentifier	-	-
Object_Name	CharacterString	STRING	“MultistateValueX”	-
Object_Type	BACnetObjectType	BACnetObjectType	MULTISTATE_VALUE	-
Present_Value	Unsigned	WORD	1	WriteProperty
Description	CharacterString	STRING	“”	WriteProperty
Status_Flags	BACnetStatusFlags	BACnetStatusFlags	‘0000’	-
Event_State	BACnetEventState	BACnetEventState	-	-
Reliability	BACnetReliability	BACnetReliability	-	WriteProperty
Out_Of_Service	BOOLEAN	BOOL	-	WriteProperty
Number_Of_States	Unsigned	WORD	1	-
State_Text	BACnetARRAY[N] of CharacterString	ARRAY [1..3] OF STRING;	“”	-
Priority_Array	BACnetPriorityArray	BACnetPriorityArray	-	-
Relinquish_Default	Unsigned	WORD	1	WriteProperty
Time_Delay	Unsigned	WORD	-	WriteProperty
Notification_Class	Unsigned	WORD	-	WriteProperty
Alarm_Values	List of Unsigned	ARRAY [1..3] OF WORD;	-	-
Fault_Values	List of Unsigned	ARRAY [1..3] OF WORD;	-	-
Event_Enable	BACnetEventTransition Bits	BACnetEventTransitionBits	‘111’	WriteProperty
Acked_Transitions	BACnetEventTransition Bits	BACnetEventTransitionBits	‘111’	-
Notify_Type	BACnetNotifyType	BACnetNotifyType	-	WriteProperty
Event Time Stamps	BACnetARRAY[3] of BACnetTimeStamp	ARRAY [1..3] OF BACnetTimeStamp;	UNSPECIFIED	-

4.12 Notification-Class-Objekt

Notification-Class-Objekte werden von anderen Objekten verwendet, die Ereignisse auslösen oder Benachrichtigungs- und Bestätigungsmethoden einsetzen.

Eine Notification-Class gibt an, wie die Handhabung von Ereignisbenachrichtigungen für TO-OFFNORMAL-, TO-FAULT- und TO-NORMAL-Ereignisse priorisiert wird, ob diese Ereigniskategorien eine Bestätigung erfordern und welche Geräte bzw. Prozesse benachrichtigt werden.

Aufgabe der Priorisierung ist es, eine Methode zur Verfügung zu stellen, mit der sichergestellt werden kann, dass zeitkritische Alarmer oder Ereignisbenachrichtigungen nicht unnötig verzögert werden. Es können Prioritäten zwischen 0 und 255 vergeben werden, eine niedrigere Zahl bedeutet eine höhere Priorität (siehe Kapitel „Prioritätsklassifizierung von Alarmen und Ereignissen“). Prioritäten können TO-OFFNORMAL-, TO-FAULT- und TO-NORMAL-Ereignissen auch einzeln ohne eine Notification-Class zugewiesen werden.

Aufgabe der Bestätigung ist es, sicherzustellen, dass die Benachrichtigung ausgeführt und nicht nur von einem anderen Gerät korrekt empfangen wurde. In den meisten Fällen erfolgt die Bestätigung durch den Anwender. TO-OFFNORMAL-, TO-FAULT- und TO-NORMAL-Ereignisse können in einer Notification-Class eine Einzelbestätigung erfordern oder nicht.

Ereignisbenachrichtigungen müssen häufig an mehrere Empfänger oder je nach Uhrzeit oder Wochentag an verschiedene Ziele gesendet werden. Notification-Class-Objekte enthalten eine Liste mit Zielen, die Uhrzeit, Wochentag und Handhabung enthalten. Ein Ziel enthält die Wochentage (Montag bis Sonntag), an denen das Notification-Class-Objekt gilt. Weiterhin enthält jedes Ziel eine FromTime und eine ToTime, die ein Zeitfenster für die Tage angeben, an denen das Ziel gültig ist. Findet ein Ereignis statt, das ein Notification-Class-Objekt verwendet, das an diesem Tag für ein bestimmtes Ziel gültig ist und die Zeit liegt im angegebenen Zeitfenster, wird eine Benachrichtigung an das Ziel gesendet. Ziele können bei Bedarf durch eine Kombination der drei Ereignisübergänge TO-OFFNORMAL, TO-FAULT und TO-NORMAL genauer spezifiziert werden.

Das Ziel gibt zudem das Gerät an, das die Benachrichtigung empfängt, sowie einen Prozess in diesem Gerät. Prozesse werden anhand numerischer Verweise identifiziert, die nur für das empfangende Gerät sinnvoll sind. Das empfangende Gerät wird entweder durch den eindeutigen Geräte-Object_Identifizier oder die BACnetAddress spezifiziert. Im zweiten Fall kann eine bestimmte Node-Adresse, eine Multicast-Adresse oder eine Broadcast-Adresse verwendet werden. Das Ziel gibt weiterhin an, ob eine zu bestätigende oder nicht zu bestätigende Ereignisbenachrichtigung gesendet wird.

Das Notification-Class-Objekt und dessen Eigenschaften werden in der folgenden Tabelle zusammengefasst.

Tabelle 15: Eigenschaften des Notification-Class-Objektes

Eigenschaften	Datentyp	Basiert auf IEC-Datentypen	Vorgabewert	Schreibbar via BACnet mittels
Object_Identifier	BACnetObjectIdentifier	BACnetObjectIdentifier	-	-
Object_Name	CharacterString	STRING	“NotificationX”	-
Object_Type	BACnetObjectType	BACnetObjectType	NOTIFICATION	-
Description	CharacterString	STRING	“”	WriteProperty
Notification_Class	Unsigned	WORD	-	-
Priority	BACnetARRAY[3] of Unsigned	ARRAY [1..3] OF WORD	127,127,127	WriteProperty
AckRequired	BACnetEventTransitionBits	BACnetEventTransitionBits	‘0000’	WriteProperty
Recipient_List	List of BACnetDestination	ARRAY [0..3] OF BACnetDestination;	-	WriteProperty

4.13 Pulse-Converter-Objekt

Das Pulse-Converter-Objekt stellt einen Prozess dar, bei dem ständige Messungen einer Menge, z. B. Strom, Wasser, Erdgas, vorgenommen und durch Impulse oder Zähler angegeben werden. Die Überwachung erfolgt beispielsweise bei der Verwaltung von Spitzenlasten über einen Zeitraum, wo regelmäßige Messungen, jedoch keine exakte Angabe jeder Eingabe oder Zählung erforderlich sind.

Das Pulse-Converter-Objekt kann einen physikalischen Eingang darstellen. Es kann auch Daten von der Eigenschaft `Present_Value` eines Accumulator-Objektes enthalten und einen Eingang desselben Gerätes wie das Pulse-Converter-Objekt darstellen. Diese Verbindung ist durch die gepunktete Linie in der folgenden Abbildung dargestellt. Bei jeder Erhöhung der Eigenschaft `Present_Value` des Accumulator-Objektes wird die Eigenschaft `Count` des Pulse-Converter-Objektes ebenfalls erhöht.

Die Eigenschaft `Present_Value` des Pulse-Converter-Objektes kann jederzeit durch Schreiben der Eigenschaft `Adjust_Value` angepasst werden. Dadurch wird die Eigenschaft `Count` angepasst und der Wert für `Present_Value` anhand dieses Wertes neu berechnet. In der Darstellung der folgenden Abbildung wurde die Eigenschaft `Count` des Pulse-Converter-Objektes auf 0 gesetzt, während der Wert für `Total_Count` des Accumulator-Objektes 0070 war.

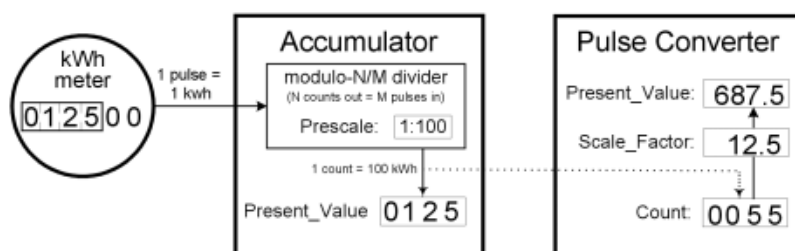


Abbildung 5: Verhältnis zwischen Pulse-Converter- und Accumulator-Objekt

Das Objekt und dessen Eigenschaften werden in der folgenden Tabelle zusammengefasst.

Tabelle 16: Eigenschaften des Pulse-Converter-Objektes

Eigenschaften	Datentyp	Basiert auf IEC-Datentypen	Vorgabewert	Schreibbar via BACnet mittels
Object_Identifier	BACnetObjectIdentifier	BACnetObjectIdentifier	-	-
Object_Name	CharacterString	STRING	“PulseConverterX”	-
Object_Type	BACnetObjectType	BACnetObjectType	PULSE_CONVERTER	-
Present_Value	REAL	REAL	-	-
Description	CharacterString	STRING	“”	WriteProperty
Input_Reference	BACnetObjectProperty-Reference	BACnetObjectPropertyReference	-	WriteProperty
Status_Flags	BACnetStatusFlags	BACnetStatusFlags	‘0000’	-
Event_State	BACnetEventState	BACnetEventState	-	-
Reliability	BACnetReliability	BACnetReliability	-	WriteProperty
Out_Of_Service	BOOLEAN	BOOL	-	WriteProperty
Units	BACnetEngineeringUnits	BACnetEngineeringUnits	None (95)	WriteProperty
Scale_Factor	REAL	REAL	1.0	WriteProperty
Adjust_Value	REAL	REAL	-	WriteProperty
Count	Unsigned	WORD	-	-
Update_Time	BACnetDateTime	BACnetDateTime	UNSPECIFIED	-
Count_Change_Time	BACnetDateTime	BACnetDateTime	UNSPECIFIED	-
Count_Before_Change	Unsigned	WORD	-	-
COV_Increment	REAL	REAL	-	WriteProperty
COV_Period	Unsigned	WORD	-	WriteProperty
Notification_Class	Unsigned	WORD	-	WriteProperty
Time_Delay	Unsigned	WORD	-	WriteProperty
High_Limit	REAL	REAL	-	WriteProperty
Low_Limit	REAL	REAL	-	WriteProperty
Deadband	REAL	REAL	-	WriteProperty
Limit_Enable	BACnetLimitEnable	BACnetLimitEnable	‘11’	WriteProperty
Event_Enable	BACnetEventTransitionBits	BACnetEventTransitionBits	‘111’	WriteProperty
Acked_Transitions	BACnetEventTransitionBits	BACnetEventTransitionBits	‘111’	-
Notify_Type	BACnetNotifyType	BACnetNotifyType	-	WriteProperty
Event Time Stamps	BACnetARRAY[3] of BACnetTimeStamp	ARRAY [1..3] OF BACnetTimeStamp	UNSPECIFIED	-

4.14 Schedule-Objekt

Das Schedule-Objekt definiert einen Zeitplan mit Ausnahmezeiten oder -zeiträumen für beliebige Tage. Das Schedule-Objekt dient auch als Verbindung zwischen diesen geplanten Zeiten und zum Schreiben festgelegter „Werte“ an bestimmte Objekteigenschaften zu diesen Zeiten. Das Schedule-Objekt und dessen Eigenschaften werden in der folgenden Tabelle zusammengefasst.

In Zeitplänen werden zwei verschiedene Arten von Tagen verwendet: Standardtage und Ausnahmetage. Beide Typen können Ereignisse für den gesamten Tag oder für einen Teil des Tages festlegen. Ein Prioritätssystem legt fest, welches geplante Ereignis zu einer bestimmten Zeit gesteuert wird.

Der aktuelle Status des Schedule-Objektes wird durch den Wert der Eigenschaft Present_Value dargestellt, der in der Regel mithilfe der Zeit/Wert-Paare der Eigenschaften Weekly_Schedule und Exception_Schedule berechnet wird. Ist kein Zeitplan aktiv, wird ein Standardwert verwendet. Details dieser Berechnung sind in der Beschreibung der Eigenschaft Present_Value enthalten.

Versionen des Schedule-Objektes vor Protocol_Revision 4 unterstützen nur Zeitpläne, die einen gesamten Tag umfassen (von Mitternacht bis Mitternacht). Zur Kompatibilität mit diesen Versionen kann das Verhalten, einen ganzen Tag zu definieren, durch Verwendung eines bestimmten Schedule-Formates erreicht werden. Weekly_Schedule- und Exception_Schedule-Werte, die mit 00:00 beginnen und keine NULL-Werte enthalten, legen Zeitpläne für einen gesamten Tag fest. Eigenschaftswerte dieses Formates führen mit allen Versionen des Schedule-Objektes zu denselben Ergebnissen.

Tabelle 17: Eigenschaften des Schedule-Objektes

Eigenschaften	Datentyp	Basiert auf IEC-Datentypen	Vorgabewert	Schreibbar via BACnet mittels
Object_Identifier	BACnetObjectIdentifier	BACnetObjectIdentifier	-	-
Object_Name	CharacterString	STRING	“ScheduleX”	-
Object_Type	BACnetObjectType	BACnetObjectType	SCHEDULE	-
Present_Value	Any	ARRAY[0..10] OF BYTE	-	-
Description	CharacterString	STRING	“”	WriteProperty
Effective_Period	BACnetDateRange	BACnetDateRange	UNSPECIFIED	WriteProperty
Weekly_Schedule	BACnetARRAY[7]of BACnetDailySchedule	ARRAY [1..7] OF BACnetDailySchedule	-	WriteProperty
Exception_Schedule	BACnetARRAY[N]of BACnetSpecialEvent	ARRAY [1..7] OF BACnetSpecialEvent	-	WriteProperty
Schedule_Default	Any	ARRAY[0..10] OF BYTE	-	WriteProperty
List_Of_Object_Property_References	List of BACnetDeviceObject-PropertyReference	ARRAY [0..3] OF BACnetDeviceObjectPropertyReference;	-	WriteProperty
Priority_For_Writing	Unsigned (1..16)	BYTE	16	WriteProperty
Status_Flags	BACnetStatusFlags	BACnetStatusFlags	‘0000’	-
Event_State				
Reliability	BACnetReliability	BACnetReliability	-	-
Out_Of_Service	BOOLEAN	BOOL	-	WriteProperty

4.15 Trend-Log-Objekt

Ein Trend-Log-Objekt überwacht eine Eigenschaft eines referenzierten Objektes. Es speichert den Wert der Eigenschaft zur späteren Verwendung mit einem Zeitstempel in einem internen Speicher, wenn vorher bestimmte Kriterien erfüllt werden. Die Daten können entweder regelmäßig oder bei Änderung eines Wertes gespeichert werden. Fehler, welche die Datenerfassung verhindern sowie Änderungen des Zustandes oder Ablaufes des Protokollvorganges selbst, werden ebenfalls gespeichert. Die Protokolleinträge werden als Trend-Log-Record bezeichnet.

Das referenzierte Objekt kann sich im selben Gerät wie das Trend-Log-Objekt oder in einem externen Gerät befinden. Der Wert der referenzierten Eigenschaft kann mit einem COV-Abonnement oder einer regelmäßigen Abfrage ermittelt werden. Sind Status-Flags verfügbar (z. B. bei Verwendung der Dienste COVNotification oder ReadPropertyMultiple), werden diese ebenfalls erfasst und mit den Daten gespeichert.

Jedem Trend-Log-Objekt wird interner Speicherplatz zugewiesen, optional mit fester Größe. Der Speicher wird gefüllt bzw. wächst, wenn Protokolleinträge hinzugefügt werden. Ist der Speicher voll, wird bei Hinzufügen eines neuen Eintrages der älteste Eintrag überschrieben oder die Datenerfassung unterbrochen. Trend-Log-Records werden mit dem Dienst ReadRange als BACnetLogRecords übertragen. Der Speicher kann durch Schreiben von Null in die Eigenschaft Record_Count gelöscht werden. Jeder Eintrag im Speicher hat eine SequenceNumber, die dem Wert der Eigenschaft Total_Record_Count zum Zeitpunkt des Hinzufügens des Eintrags entspricht. Ist der Wert von Total_Record_Count größer als 232-1, wird die Eigenschaft auf „1“ zurückgesetzt.

Die Protokollierung kann mithilfe der Eigenschaft Log_Enable und durch in den Eigenschaften Start_Time und Stop_Time angegebene Daten und Zeiten aktiviert/deaktiviert werden. Die Aktivierung/Deaktivierung des Trend-Log wird im Protokollspeicher gespeichert.

Für das automatische Abholen von Protokolleinträgen durch Prozesse oder andere Geräte wie Dateiserver kann die Ereignisbenachrichtigung verwendet werden.

Bei dem objektinternen Melden wird eine neue Benachrichtigung an alle im Abonnement enthaltenen Geräte gesendet, wenn seit Versenden der letzten Benachrichtigung bzw. seit dem Start die in der Eigenschaft Notification_Threshold angegebene Anzahl an Protokolleinträgen erreicht wird.

Als Reaktion auf eine Benachrichtigung können Abonnenten alle neuen Protokolleinträge abholen. Hierfür sollte die Methode „By Sequence Number“ des Dienstes ReadRange verwendet werden.

Ein Abonnent erkennt eine verpasste Benachrichtigung daran, dass der Parameter Current_Notify_Record der letzten Benachrichtigung nicht dem Parameter Previous_Notify_Record der aktuellen Benachrichtigung entspricht. Ist bei der in diesem Fall erhaltenen Antwort ReadRange-ACK auf die ReadRange-Anfrage das

First-Item-Flag auf „TRUE“ gesetzt, hat dieser Abonnent wahrscheinlich Trend-Log-Records verpasst.

Der Empfang von Protokolleinträgen durch entfernte Geräte hat keinen Einfluss auf den Zustand des Trend-Log-Objektes selbst. Dies ermöglicht einen unabhängigen Zugriff auf Protokolleinträge in korrekter Reihenfolge durch alle entfernten Geräte. Jedes entfernte Gerät kann seine Einträge jederzeit aktualisieren.

Hinweis**Bedingungen für dauerhaftes Speichern von Trending-Werten beachten!**

Trending-Werte werden zunächst im RAM gesichert.

Das dauerhafte Speichern im Flash erfolgt unter folgenden Bedingungen:

- 30 Minuten nach dem ersten ungespeicherten Wert
- wenn der Log_Buffer voll ist, in welchem die Records gespeichert werden (die Eigenschaft "Stop_When_Full" ist auf TRUE gesetzt)
- wenn das Trend-Log-Objekt Intrinsic-Reporting unterstützt und durch das Hinzufügen eines Records zum Cache die Intrinsic-Reporting-Bedingung (Records_Since_Notification $\geq$ Notification_Threshold) zutrifft
- wenn Sie die Eigenschaft "Log_Enable" des Trend-Log-Objektes deaktivieren
- wenn Sie einen Software-Reset über BACnet durchführen

Im Falle eines ungeplanten Resets können die Werte von maximal 30 Minuten verloren gehen.

Tabelle 18: Eigenschaften des Trend-Log-Objektes

Eigenschaften	Datentyp	Basiert auf IEC-Datentypen	Vorgabewert	Schreibbar via BACnet mittels
Object_Identifier	BACnetObjectIdentifier	BACnetObjectIdentifier	-	-
Object_Name	CharacterString	STRING	“TrendlogX”	-
Object_Type	BACnetObjectType	BACnetObjectType	TRENDLOG	-
Description	CharacterString	STRING	“”	WriteProperty
Log_Enable	BOOLEAN	BOOL	-	WriteProperty
Start_Time	BACnetDateTime	BACnetDateTime	UNSPECIFIED	WriteProperty
Stop_Time	BACnetDateTime	BACnetDateTime	UNSPECIFIED	WriteProperty
Log_DeviceObjectProperty	BACnetDeviceObject-PropertyReference	BACnetDeviceObject-PropertyReference	-	WriteProperty
Log_Interval	Unsigned	WORD	-	WriteProperty
COV_Resubscription_Interval	Unsigned	WORD	-	WriteProperty
Client_COV_Increment	BACnetClientCOV	BACnetClientCOV	-	WriteProperty
Stop_When_Full	BOOLEAN	BOOL	-	WriteProperty
Buffer_Size	Unsigned32	DWORD	-	-
Log_Buffer	List of BACnetLogRecord	ARRAY [0..3] OF BACnetLogRecord;	-	-
Record_Count	Unsigned32	DWORD	-	WriteProperty
Total_Record_Count	Unsigned32	DWORD	-	-
Notification_Threshold	Unsigned32	DWORD	-	WriteProperty
Records_Since_Notification	Unsigned32	DWORD		-
Last_Notify_Record	Unsigned32	DWORD	-	-
Event_State	BACnetEventState	BACnetEventState	-	-
Notification_Class	Unsigned	WORD	-	WriteProperty
Event_Enable	BACnetEventTransitionBits	BACnetEventTransitionBits	‘111’	WriteProperty
Acked_Transitions	BACnetEventTransitionBits	BACnetEventTransitionBits	‘111’	-
Notify_Type	BACnetNotifyType	BACnetNotifyType	-	WriteProperty
Event Time Stamps	BACnetARRAY[3] of BACnetTimeStamp	ARRAY [1..3] OF BACnetTimeStamp	UNSPECIFIED	-

5 BACnet-Dienste

5.1 Alarm- und Ereignisdienste

Dieser Abschnitt beschreibt das BACnet-Konzept und die BACnet-Dienste zur Behandlung und Verwaltung von Ereignissen. Objekte der Ereignisverwaltung werden in Kapitel „Objekte im nicht-nativen Betrieb“ definiert. Im Allgemeinen ist ein Ereignis die Änderung des Wertes bestimmter Eigenschaften von bestimmten Objekten oder eine interne Zustandsänderungen, die vorher bestimmte Kriterien erfüllt.

Es gibt drei BACnet-Methoden zur Ereignisverwaltung (siehe folgende Kapitel):

- Melden bei Wertänderungen („Change Of Value“/„State Reporting“)
- Objektinternes Melden („Intrinsic Reporting“)
- Regelbasiertes Melden („Algorithmic Change Reporting“)

Bei **regelbasiertem Melden** kann ein Client einen Benachrichtigungsserver abonnieren, um Berichte über Werteänderungen beliebiger Eigenschaften beliebiger Objekte auf Grundlage vorher bestimmter und im Netzwerk sichtbarer Kriterien erhalten. Jede Standardregel kann für die Erstellung von Kriterien für Benachrichtigungen verwendet werden. Nach der Erstellung werden bei Änderungen Benachrichtigungen an ein oder mehrere Ziele auf Grundlage der Kriterien versandt. Regelbasiertes Melden unterscheidet sich vom objektinternen Melden dadurch, dass für die Bestimmung des Ereigniszustandes Event-Enrollment-Objekte verwendet werden.

Hinweis



Regelbasierte Meldungen werden nicht unterstützt!

Der BACnet/IP-Controller 750-830 unterstützt keine regelbasierten Meldungen.

Objektinternes und regelbasiertes Melden basieren auf dem Konzept der Ereignisklassifizierung für gezielte Berichte einzelner Ereignisse. Bestimmte objektinterne oder regelbasierte Ereignisse können einen Notification-Class-Wert angeben, der in direktem Zusammenhang mit einem Notification-Class-Objekt des entsprechenden Gerätes steht. Das Notification-Class-Objekt wird für Angaben zu Handhabung und Routing der Ereignisse an ein oder mehrere Ziele verwendet. Das Notification-Class-Objekt gibt die Prioritäten für Ereignisbenachrichtigungen an, ob eine Bestätigung durch einen Anwendungsprozess oder einen Anwender erforderlich ist und zu welchen Zeiträumen in der Woche welche Ziele verwendet werden.

BACnet-Geräte, die Ereignisbenachrichtigungen unterstützen, können beliebig viele Einzelbedingungen definieren, die Alarm- oder Ereignisbenachrichtigungen auslösen. Alarme und Ereignisse werden grob in drei Gruppen eingeteilt: **FAULT**, **OFFNORMAL** und **NORMAL**.

- Der Zustand **FAULT** ist eine Fehlfunktion, in der Regel eine Störung im Automatisierungssystem selbst.
- Der Zustand **OFFNORMAL** ist ein Systemzustand, der unerwartet ist oder außerhalb der Grenzwerte für einen optimalen Betrieb liegt.
- Jeder andere Zustand ist **NORMAL**.

Objekte, die Ereignisse auslösen, können jederzeit einen von beliebig vielen eindeutigen Ereigniszuständen annehmen. Benachrichtigungen werden durch Zustandsänderungen eines Objektes ausgelöst, in der Regel von einem eindeutigen Ereigniszustand in einen anderen. Der Wechsel in einen bestimmten Ereigniszustand kann für die Bestimmung einer eindeutigen Handhabung der durch ein Objekt generierten Benachrichtigungen verwendet werden, beispielsweise für die Bestimmung des Benachrichtigungsziels oder ob ein bestimmtes Änderungsereignis eine Bestätigung erfordert. In diesem Kontext werden alle Änderungen, die zu einem **NON-NORMAL**- oder **FAULT**-Zustand führen, als **TO-OFFNORMAL**-Änderungen angesehen. Änderungen in einen **FAULT**-Zustand werden als **TO-FAULT**-Änderungen angesehen. Alle anderen Änderungen sind als **TO-NORMAL**-Änderungen definiert.

Ereignisse können einzeln als Alarm oder Ereignis erkannt werden. Objekte, die Ereignisse auslösen, geben diesen Unterschied mithilfe der Eigenschaft **Notify_Type** an. In BACnet liegt der einzige Unterschied zwischen Alarmen und allen anderen Ereignissen darin, dass der Dienst **GetAlarmSummary** nur Alarme zurückgibt und keine Ereignisse. In allen anderen Belangen unterscheidet BACnet nicht zwischen Alarmen und Ereignissen.

5.1.1 Melden bei Wertänderungen (Change Of Value / State Reporting)

Beim Melden von Wertänderungen kann ein Client temporär einen COV-Server abonnieren, um Berichte über bestimmte Werteänderungen der referenzierten Eigenschaften auf Grundlage bestimmter Kriterien zu erhalten. Einige BACnet-Standardobjekte unterstützen COV-Reporting optional. Unterstützt ein Objekt COV-Reporting, lösen Werteänderungen abonniertter Eigenschaften des Objektes (basierend auf programmierbaren Schwellenwerten) das Senden von COV-Benachrichtigungen an einen oder mehrere Clients mit Abonnement aus. In der Regel werden COV-Benachrichtigungen an Überwachungsprogramme in COV-Client-Geräten, an Anwender oder an Protokollgeräte gesendet.

COV-Abonnements werden mit einem der Dienste `SubscribeCOV` oder `SubscribeCOVProperty` erstellt. Das Abonnement erstellt eine Verbindung zwischen Erkennung der Werteänderung und der Reporting-Methode des COV-Server-Gerätes sowie einen Prozess im COV-Client-Gerät.

Änderungsbenachrichtigungen werden vom COV-Server ausgegeben, wenn nach Erstellung des Abonnements Änderungen auftreten. Der COV-Server verwendet die Dienste `ConfirmedCOVNotification` und `UnconfirmedCOVNotification`, um die Änderungsbenachrichtigung zu übermitteln. Der Versand zu bestätigender bzw. nicht zu bestätigender Nachrichten wird im Abonnement angegeben.

Unterstützt ein BACnet-Standardobjekt eines in der folgenden Tabelle aufgeführten Typs COV-Reporting, muss die Unterstützung für COV-Reporting für die Eigenschaft entsprechend dieser Tabelle erfolgen. Bei den aufgeführten Eigenschaften des Datentyps REAL entspricht der COV-Schwellenwert zur Bestimmung, ob eine Benachrichtigung generiert wird oder nicht, der Eigenschaft `COV_Increment` des Objektes, außer der Parameter `COV_Increment` wird durch den Dienst `SubscribeCOVProperty` zur Verfügung gestellt. Bei anderen Eigenschaften des Datentyps REAL ist der COV-Schwellenwert lokal unterschiedlich, wenn er nicht durch den Dienst `SubscribeCOVProperty` zur Verfügung gestellt wird. So können mehrere Abonnenten, die keinen bestimmten Schwellenwert erfordern, einen gemeinsamen Schwellenwert verwenden. Dadurch wird der COV-Server weniger belastet. Die Kriterien für COV-Reporting bei anderen als in der folgenden Tabelle aufgeführten Eigenschaften basiert auf dem Datentyp der Eigenschaft mit Abonnement und ist in der darauf folgenden Tabelle aufgeführt.

Unterstützt ein Objekt die Eigenschaft `COV_Period` und ist diese nicht Null, werden in dem Intervall, welches in `COV_Period` angegeben ist, regelmäßig COV-Benachrichtigungen an alle Empfänger mit Abonnement versandt. Der Versand erfolgt zusätzlich zu den bei Werteänderungen überwachter Eigenschaften versandten Benachrichtigungen. Der durch regelmäßige COV-Benachrichtigungen übertragene Wert der überwachten Eigenschaft ist die Grundlage für die Ermittlung, ob eine weitere COV-Benachrichtigung für die Werteänderung der überwachten Eigenschaft erforderlich ist.

Ist `COV_Period` = Null, werden keine regelmäßigen Benachrichtigungen versandt.

Die Liste mit Abonnements ist mithilfe der Eigenschaft `Active_COV_Subscriptions` des Device-Objektes im Netzwerk sichtbar. Abonnements können mit begrenzter Lebensdauer erstellt werden, d. h. das Abonnement läuft nach einer bestimmten Zeit automatisch ab. Optional kann eine unbegrenzte Lebensdauer angegeben werden, d. h. das Abonnement läuft nicht automatisch ab.

VORSICHT **Keine Garantie für Abonnements nach Stromausfall oder Neustart!**



Der COV-Server garantiert keinen Erhalt der Abonnements bei einem Stromausfall oder Neustart.

Regelmäßige Erneuerung von Abonnements ist zulässig und wird erwartet, ein erneuertes Abonnement wird als neues Abonnement behandelt und verlängert die Lebensdauer des Abonnements.

Verschiedene Standardobjekte, die COV-Reporting unterstützen, verwenden verschiedene Kriterien für die Bestimmung, ob eine Werteänderung stattgefunden hat, siehe folgende Tabelle. Proprietäre Objekte oder andere Standardobjekte als in dieser Tabelle aufgeführt, die COV-Reporting der Eigenschaft `Present_Value` unterstützen, sollten diese Kriterien falls möglich erfüllen. Objekte, die COV optional unterstützen, sowie die zu verwendenden COV-Algorithmen sind in den folgenden beiden Tabellen aufgeführt.

Tabelle 19: Standardisierte Objekte, die unter Umständen COV-Reporting unterstützen

Objekt	Kriterien	Eigenschaften für Reporting
Analog Input, Analog Output, Analog-Value	Bei Änderung von Present_Value um COV_Increment oder bei Änderung von Status_Flags	Present_Value, Status_Flags
Binary-Input, Binary-Output, Binary-Value, Life-Safety-Point, Life-Safety-Zone, Multi-State-Input, Multi-State-Output, Multi-State-Value	Bei Änderung von Present_Value oder bei Änderung von Status_Flags	Present_Value, Status_Flags
Pulse-Converter	Bei Änderung von Present_Value um COV_Increment oder bei Änderung von Status_Flags oder bei Ablauf von COV_Period	Present_Value, Status_Flags, Update_Time

Tabelle 20: Kriterien für die Verwendung von COV-Reporting mit anderen Eigenschaften als in
voriger Tabelle aufgeführt

Datentyp	Kriterien	Eigenschaften für Reporting
REAL	Bei Erhöhen des Wertes der Eigenschaft um den Schwellenwert (falls verfügbar vom Dienst, sonst wie vom Gerät bestimmt) oder bei Änderungen von Status_Flags (falls das Objekt über die Eigenschaft Status_Flags verfügt)	Die abonnierte Eigenschaft Status_Flags (falls das Objekt über die Eigenschaft Status_Flags verfügt)
Alle anderen Datentypen	Bei Änderungen der Eigenschaft oder bei Änderungen von Status_Flags (falls das Objekt über die Eigenschaft Status_Flags verfügt)	Die abonnierte Eigenschaft Status_Flags (falls das Objekt über die Eigenschaft Status_Flags verfügt)

5.1.2 Objektinternes Melden

Bestimmte BACnet-Standardobjekte unterstützen objektinternes Melden durch optionale Eigenschaften zur Definition des zu erstellenden Alarm- oder Ereignistyps sowie für Handhabung und Routing der Benachrichtigungen. Interne Statusänderungen und Alarmer können objektinternes Melden für die Generierung von Diagnosebenachrichtigungen verwenden.

Unterstützt ein Standardobjekt objektinternes Melden, lösen Werteänderungen bestimmter Eigenschaften des Objektes (in einigen Fällen basierend auf programmierbaren Kriterien) oder Änderungen des internen Zustandes des Objektes das Senden von Benachrichtigungen aus. Diese werden an ein oder mehrere Ziele basierend auf der Notification-Class gesendet. In der Regel werden Ereignisbenachrichtigungen an Anwender oder an Protokollgeräte gesendet, die im Client-Gerät als Prozesse dargestellt werden. Die Dienste `ConfirmedEventNotification` und `UnconfirmedEventNotification` werden vom Benachrichtigungsserver für die Nachrichtenübergabe verwendet.

Die Verbindung zwischen der Methode für Ereigniserkennung und Reporting des Gerätes, welche die Ereignisbenachrichtigung initiiert, und den Prozessen in einem oder mehreren Client-Geräten zur Benachrichtigung erfolgt indirekt durch ein Notification-Class-Objekt im initiiierenden Gerät. Das Notification-Class-Objekt kann in Geräten statisch oder dynamisch konfiguriert werden, wenn die Geräte dynamische Änderung oder Erstellung von Notification-Class-Objekten unterstützen. Mehrere Objekte, die Alarmer oder Ereignisse initiieren, können auf dasselbe Notification-Class-Objekt eines Gerätes verweisen. Die Notification-Class bietet folgende Informationen zur Handhabung einer Benachrichtigung:

- Angabe der Priorität für die Ereignisbenachrichtigung, basierend auf dem Typ der Zustandsänderung (TO-OFFNORMAL, TO-FAULT oder TO-NORMAL)
- Angabe, ob das Ereignis eine Bestätigung erfordert, basierend auf dem Typ der Zustandsänderung
- Angabe eines bestimmten Ziels für die Ereignisbenachrichtigung (Prozess, Empfänger) als Tupel, basierend auf Uhrzeit, Wochentag und Typ der Zustandsänderung
- Angabe, ob an ein bestimmtes Ziel zu bestätigende oder nicht zu bestätigende Benachrichtigungen gesendet werden

Verschiedene Objekte verwenden unterschiedliche standardisierte Kriterien für die Bestimmung, ob ein Ereignis stattgefunden hat. Standardisierte Objekte, die objektinterne Ereignisbenachrichtigungen optional unterstützen sowie die hierbei verwendeten Ereignistypen sind in zusammengefasst. Die Objekteigenschaften werden in Kapitel „Objekte im nicht-nativen Betrieb“ beschrieben.

Die in folgender Tabelle aufgeführten Standard-Objekte verwenden Standard-Ereignistypen für das Melden von zu Ereignissen gehörenden Parameterwerten. Diese Werte werden im Parameter „Event Values“ der Dienste

ConfirmedEventNotification und UnconfirmedEventNotification zurückgegeben. Der Ereignistyp bestimmt, welcher Wertesatz für jeden Standard-Ereignistyp zurückgegeben wird. Die zurückgegebenen Werte sind in der darauf folgenden Tabelle zusammengefasst. Proprietäre Objekte, die keinen der Standard-Objekte verwenden, können stattdessen eine Liste mit Eigenschaftswerten zurückgeben.

Bei Binary-Input- und Binary-Value-Objekten gibt die Eigenschaft Alarm_Value an, welcher der beiden Zustände von Present_Value als OFFNORMAL gilt. Der andere Zustand gilt als NORMAL. Beim Wechsel in einen oder beide Zustände können Benachrichtigungen generiert werden, wenn die entsprechenden Flags in Event_Enable gesetzt sind.

Bei Multi-State-Value-, Life-Safety-Point- und Life-Safety-Zone-Objekten enthält die Eigenschaft Alarm_Values jeden möglichen Zustand von Present_Value, der als OFFNORMAL gilt. Die Eigenschaft Fault_Values führt jeden möglichen Zustand von Present_Value auf, der als FAULT gilt. Alle anderen Zustände von Present_Value gelten als NORMAL. Beim Wechsel in einen der Zustände können Benachrichtigungen generiert werden, wenn die entsprechenden Flags in Event_Enable gesetzt sind.

Bei Life-Safety-Point- und Life-Safety-Zone-Objekten enthält die Eigenschaft Life_Safety_Alarm_Values jeden möglichen Zustand von Present_Value, der als LIFE_SAFETY_ALARM gilt.

Tabelle 21: Standardisierte Objekte, die objektinternes Melden unterstützen

Objekt	Kriterien	Ereignistyp
Binary-Input, Binary-Value, Multi-State-Value	Bei Änderung von Present_Value in einen neuen Zustand für länger als in Time_Delay angegeben und die Änderung ist in Event_Enable aktiviert	CHANGE_OF_STATE
Binary-Output	Bei Unterschied zwischen Present_Value und Feedback_Value für länger als in Time_Delay angegeben und die Änderung ist in Event_Enable aktiviert	COMMAND_FAILURE
Analog Input, Analog Output, Analog-Value, Pulse-Converter	Wenn Present_Value den von High_Limit und Low_Limit vorgegebenen Bereich für länger als in Time_Delay angegeben übersteigt und der Wechsel in Event_Enable sowie Limit_Enable aktiviert ist und Present_Value kehrt in den von High_Limit - Deadband und Low_Limit + Deadband vorgegebenen Bereich für länger als in Time_Delay angegeben zurück und der Wechsel ist in Event_Enable sowie Limit_Enable aktiviert	OUT_OF_RANGE
Trend-Log	Wenn der Event_State NORMAL und Records_Since_Notification gleich dem Notification_Threshold ist	BUFFER_READY

Tabelle 21: Standardisierte Objekte, die objektinternes Melden unterstützen

Objekt	Kriterien	Ereignistyp
Life-Safety-Point, Life-Safety-Zone	Wenn Present_Value auf einen der Werte in der Liste Life_Safety_Alarm_Values geändert wird und länger als in Time_Delay angegeben gleich einem der Werte in der Liste Life_Safety_Alarm_Values und der Wechsel in Event-Enable aktiviert ist oder wenn Present_Value auf einen der Werte in der Liste Alarm_Values geändert wird und länger als in Time_Delay angegeben gleich einem der Werte in der Liste Alarm_Values ist und der Wechsel in Event-Enable aktiviert ist oder bei Modusänderungen	CHANGE_OF_LIFE_SAFETY
Accumulator	Wenn Pulse_Rate den von High_Limit und Low_Limit vorgegebenen Bereich für länger als in Time_Delay angegeben übersteigt und der Wechsel in Event_Enable sowie Limit_Enable aktiviert ist oder Pulse_Rate kehrt in den von High_Limit und Low_Limit vorgegebenen Bereich für länger als in Time_Delay angegeben zurück und der Wechsel ist in Event_Enable sowie Limit_Enable aktiviert	UNSIGNED_RANGE

Tabelle 22: : Standard-Objekteigenschaftswerte, die in Benachrichtigungen zurückgegeben werden.

Objekt	Ereignistyp	Benachrichtigungsparameter	Eigenschaften des referenzierten Objektes
Binary-Input, Binary-Value, Multi-State-Value	CHANGE_OF_STATE	New_State Status_Flags	Present_Value Status_Flags
Binary-Output	COMMAND_FAILURE	Command_Value Status_Flags Feedback_Value	Present_Value Status_Flags Feedback_Value
Analog Input, Analog Output, Analog-Value, Pulse-Converter	OUT_OF_RANGE	Exceeding_Value Status_Flags Deadband Exceeded_Limit	Present_Value Status_Flags Deadband Low_Limit oder High_Limit
Trand-Log	BUFFER_READY	Buffer_Property Previous_Notification Current_Notification	BACnetDevice ObjectProperty Reference Last_Notify_Record Total_Record_Count
Life-Safety-Point, Life-Safety-Zone	CHANGE_OF_LIFE_ SAFETY	New_State New_Mode Status_Flags Operation_Expected	Present_Value Mode Status_Flags Operation_Expected
Accumulator	UNSIGNED_RANGE	Exceeding_Value Status_Flags Exceeded_Limit	Pulse_Rate Status_Flags Low_Limit oder High_Limit

Tabelle 23: Benachrichtigungsparameter für Standard-Ereignistypen

Ereignistyp	Benachrichtigungsparameter	Beschreibung
CHANGE_OF_BITSTRING	Referenced_Bitstring	Der neue Wert der referenzierten Bitstring-Eigenschaft
	Status_Flags	Status_Flags des referenzierten Objektes
CHANGE_OF_STATE	New_State	Der neue Wert der referenzierten Eigenschaft
	Status_Flags	Status_Flags des referenzierten Objektes
CHANGE_OF_VALUE	New_Value	Der neue Wert der referenzierten Eigenschaft
COMMAND_FAILURE	Command_Value	Der Wert der Eigenschaft, die den Befehl empfängt.
	Status_Flags	Status_Flags des referenzierten Objektes.
	Feedback_Value	Der Wert, der sich vom Command_Value unterscheidet
FLOATING_LIMIT	Referenced_Value	Der neue Wert der referenzierten Eigenschaft
	Status_Flags	Status_Flags des referenzierten Objektes
	Setpoint_Value	Der Wert des Setpoint-Verweises
	Error_Limit	Der Unterschieds-Grenzwert, der überschritten wurde
FLOATING_LIMIT	Referenced_Value	Der neue Wert der referenzierten Eigenschaft
	Status_Flags	Status_Flags des referenzierten Objektes
	Setpoint_Value	Der Wert des Setpoint-Verweises
	Error_Limit	Der Unterschieds-Grenzwert, der überschritten wurde

Tabelle 23: Benachrichtigungsparameter für Standard-Ereignistypen

Ereignistyp	Benachrichtigungsparameter	Beschreibung
OUT_OF_RANGE	Exceeding_Value	Der Werte hat einen Grenzwert überschritten
	Status_Flags	Status_Flags des referenzierten Objektes
	Deadband	Für Überprüfung des Grenzwert verwendetes Deadband
	Exceeded_Limit	Der überschrittene Grenzwert
BUFFER_READY	Buffer_Property	Verweist auf die Buffer-Eigenschaft
	Previous_Notification	Der Parameter Current_Notification der vorherigen Benachrichtigung oder „0“, falls keine vorherige Benachrichtigung gesendet wurde
	Current_Notification	SequenceNumber des Eintrags, der diese Benachrichtigung ausgelöst hat
CHANGE_OF_LIFE_SAFETY	New_State	Der neue Wert der referenzierten Eigenschaft
	New_Mode	Der neue Mode des referenzierten Objektes
	Status_Flags	Status_Flags des referenzierten Objektes
	Operation_Expected	Die nächste vom referenzierten Objekt angefragte Aktion
UNSIGNED_RANGE	Exceeding_Value	Der einen Grenzwert übersteigende Wert
	Status_Flags	Status_Flags des referenzierten Objektes
	Exceeded_Limit	Der Überschrittene Grenzwert

5.1.3 Auftreten und Benachrichtigung von Alarmen und Ereignissen

Die Verbindung zwischen Auftreten eines Ereignisses und Übertragung der Ereignisbenachrichtigung an einen oder mehrere Empfänger kann auf verschiedene Arten hergestellt werden, je nach gewünschter Reporting-Methode. COV-Ereignisse sind über den Dienst SubscribeCOV mit einem oder mehreren Abonnenten verbunden. Objektinterne Ereignisse sind indirekt durch Verknüpfung mit einem Notification-Class-Objekt mit einem oder mehreren Empfängern verbunden.

Die Kriterien für die Ereigniserkennung werden durch Parameter des Dienstes SubscribeCOV und durch Eigenschaften der Objekte angegeben, die Ereignisbenachrichtigungen generieren können. Eine Beschreibung dieser Eigenschaften ist bei der Beschreibung der entsprechenden Eigenschaften enthalten, siehe Tabelle „Standardisierte Objekte, die objektinternes Melden unterstützen“ in Kapitel „Objektinternes Melden“. Alle Objekte, die Ereignisse generieren, verfügen über eine Eigenschaft, die den Zustand des Objektes in Bezug auf Alarm- und Ereignishandhabung angibt. Für jedes dieser Objekte kann die Meldung bestimmter Ereignisse explizit aktiviert oder deaktiviert werden.

COV- und Ereignisbenachrichtigungen können Benachrichtigungen versenden, die eine Bestätigung erfordern oder nicht. Die Wahl nach der zu verwendenden Benachrichtigungsmethode richtet sich nach der Bedeutung jedes Ereignisses und ob eine Benachrichtigung erforderlich oder lediglich wünschenswert ist. Im ersten Fall kann eine Benachrichtigung, die eine Bestätigung erfordert, an beliebig viele Empfänger versandt werden. Im zweiten Fall kann eine Benachrichtigung, die keine Bestätigung erfordert, an eine Multicast- oder eine Broadcast-Adresse versandt werden.

Notification-Class-Objekte geben das Zielgerät für Benachrichtigungen mit „BACnetRecipients“ an. Die Empfänger sind einzelne Geräte, Gerätegruppen mit einer gemeinsamen Multicast-Adresse oder alle Geräte, die über eine Broadcast-Adresse erreicht werden können. Bei Verwendung von Broadcast ist der Umfang unter Umständen auf Geräte eines Netzwerkes beschränkt oder kann auf alle Geräte eines BACnet-Netzwerkes erweitert werden.

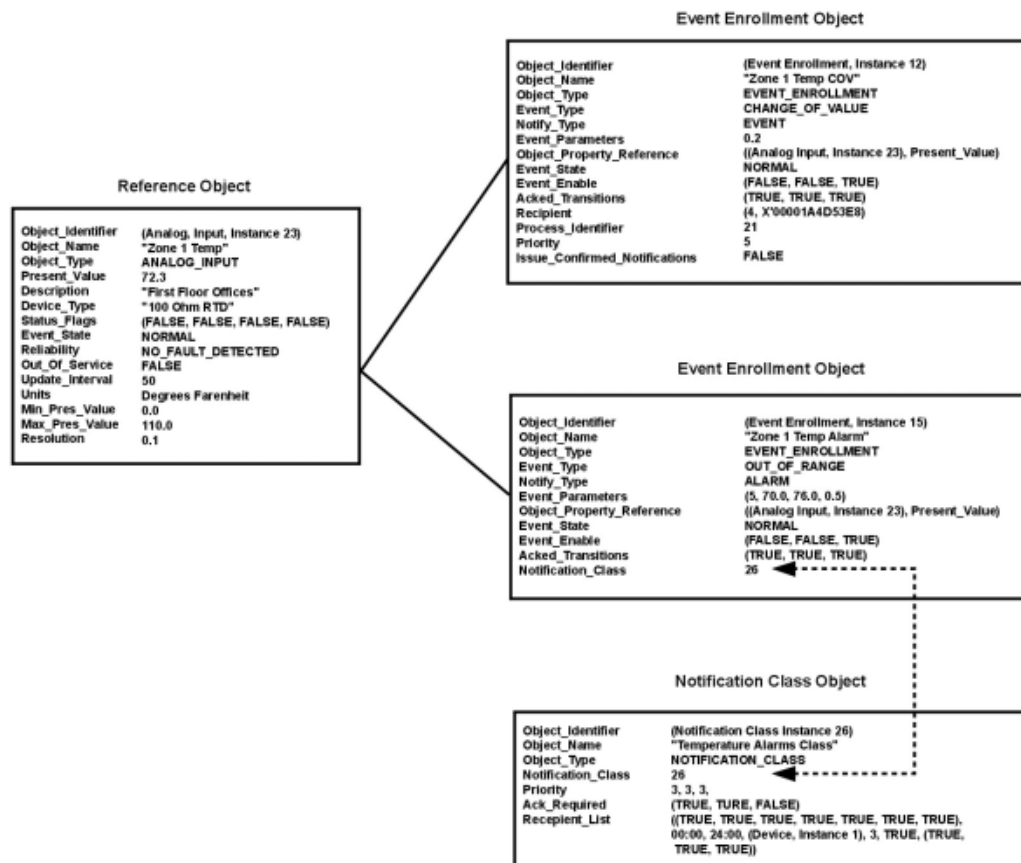


Abbildung 6: Beispiel für Event-Enrollment

Die Dienste zur Ereignisbenachrichtigung enthalten einen Parameter „Time Stamp“, der die chronologische Reihenfolge von Ereignissen angibt. Der „Time Stamp“ kann die aktuelle, durch den Takt des lokalen Gerätes ermittelte, Zeit sein oder, falls das Gerät nicht getaktet ist, eine Sequenznummer. Sequenznummern steigen stetig bis zum Maximalwert, anschließend fängt die Zählung wieder bei Null an. Ein Gerät kann über eine einzelne Sequenznummer für alle Objekte verfügen, die Ereignisse auslösen, oder es kann für jedes Objekt eine eigene Sequenznummer verwendet werden.

Für die Ereignisverwaltung sind elf Dienste definiert:

- AcknowledgeAlarm
- ConfirmedCOVNotification
- UnconfirmedCOVNotification
- ConfirmedEventNotification
- UnconfirmedEventNotification
- GetAlarmSummary
- GetEnrollmentSummary
- GetEventInformation
- LifeSafetyOperation
- SubscribeCOV
- SubscribeCOVProperty

5.1.3.1 Prioritätsklassifizierung von Alarmen und Ereignissen

Alarme und Ereignisse in einem BACnet-Netzwerk erfordern eine Priorisierung (siehe folgende Tabelle), um sicherzustellen, dass wichtige Informationen schnell bearbeitet werden und ihr Ziel erreichen.

Tabelle 24: Alarm- und Ereignispriorität - Zuweisung von Netzwerkpriorität

Alarm- und Ereignispriorität	Netzwerkpriorität
00 – 63	Life-Safety-Meldung
64 – 127	Critical-Equipment-Meldung
128 – 191	Urgent-Meldung
192 – 255	Normal-Meldung

5.1.4 Terminologie der Darstellung

Die folgenden Kapitel beinhalten Dienstabellen, die nach einem bestimmten Muster aufgebaut sind. Es werden Interaktionen zwischen einem Dienst-Nutzer und einem Dienst-Bereitsteller dargestellt. Jeder Dienst hat eine Reihe von Null oder mehr Parametern, die auf Anforderung übertragen werden.

In den folgenden Kapiteln wird ein Tabulator-Format zur Beschreibung der BACnet-Dienst-Parameter verwendet. Jede Tabelle besteht aus fünf Spalten, die den Namen der Dienst-Parameter sowie die Dienst-Primitiven Anfrage („Req“), Anzeige („Ind“), Antwort („Rsp“) und Bestätigung („Cnf“) enthalten. Die Spalten „Rsp“ und „Cnf“ stehen für unbestätigte Dienste nicht zur Verfügung. Jede Zeile der Tabelle enthält einen Parameter oder Subparameter. Unter den entsprechenden Spalten der Dienst-Primitive wird ein Code verwendet, um die Art der Benutzung des Dienst-Parameters zu spezifizieren, (senkrechten Spalte):

- M** Der Parameter ist für den Dienst vorgeschrieben.
- U** Der Parameter ist eine Benutzeroption und darf nicht zur Verfügung gestellt werden.
- C** Der Parameter ist durch andere Parameter bedingt.
- S** Der Parameter ist eine Auswahl aus einer Sammlung von zwei oder mehreren möglichen Parametern. Die Parameter, die diese Sammlung umfasst, werden folgendermaßen in der Tabelle angegeben:
 - (a) Jeder Parameter in der Sammlung wird durch den Code „S“ spezifiziert.
 - (b) Der Name jedes Parameters in der Sammlung befindet sich in derselben Tabellenmarkierung, am Anfang der Parameter-Spalte in der Tabelle.
 - (c) Entweder jeder Parameter befindet sich in der Kategorie links in der Tabelle oder jeder Parameter ist Teil ein und derselben Parametergruppe.
Eine Parametergruppe ist eine Sammlung von Parametern, wobei jedes Mitglied einen gemeinsamen Stammparameter hat.
Der Stammparameter für jedes Gruppenmitglied ist der erste Parameter über dem Mitglied, der weiter links eingerückt ist.

In dem folgenden Beispiel bilden ParameterA und ParameterB eine Parametergruppe:

```
StammparameterX
    ParameterA
    ParameterB
StammparameterY
    ParameterC
```

Parameter einer Sammlung werden in den Dienstabellen durch Einrückung gekennzeichnet. Alle Parameter mit derselben Einrückung und einem gemeinsamen Stammparameter sind Teil derselben Auswahl.

Die Zeichen „(=)“, die nach einem der Codes „M“, „U“, „C“ oder „S“ stehen, zeigen an, dass der Parameter semantisch dem Parameter in der Dienst-Primitive entspricht, der links neben ihm in der Tabelle steht. Beispiel: Ein Code „M(=)“ in der Spalte „Ind“ und „M“ in der Spalte „Req“ bedeutet, dass der Parameter unter „Ind“ semantisch dem Parameter der Anfrage „Req“ entspricht.

Einige Parameter können Subparameter enthalten. Subparameter werden durch Einrückung gegenüber dem Stammparameter angezeigt. Die Präsenz von Subparametern hängt immer von der Präsenz des Stammparameters ab. In dem oben aufgeführten Beispiel sind die ParameterA und ParameterB Subparameter von StammparameterX, und ParameterC ist ein Subparameter von StammparameterY. Wenn StammparameterX optional ist und nicht in einer Dienst-Primitive bereitgestellt wird, dürfen die Subparameter (ParameterA und ParameterB) nicht bereitgestellt werden.

Einige Dienstparameter werden als „List of ...“ dargestellt. Sofern nicht anders angegeben, spezifizieren alle Parameter, deren Name mit „List of ...“ beginnen, eine Liste mit keinem oder mehreren Objekten.

Die Parameter werden in der erläuterten Terminologie nachfolgend tabellarisch für jeden Dienst aufgeführt. Eine ausführliche Beschreibung jedes Parameters entnehmen Sie Kapitel „Gesamtliste der BACnet-Dienst-Parameter“.

5.1.5 AcknowledgeAlarm-Dienst

In einigen Systemen muss ein Gerät darüber informiert werden, ob ein Anwender eine Alarmmeldung gesehen hat. Der Dienst AcknowledgeAlarm wird vom Benachrichtigungs-Client verwendet, um zu bestätigen, dass ein Anwender eine Ereignisbenachrichtigung mit „AckRequired = TRUE“ gesehen und bestätigt hat. Die Gewährleistung, dass eine Bestätigung tatsächlich von einem Anwender mit der entsprechenden Autorisierung stammt, ist lokal unterschiedlich. Der Dienst kann in Verbindung mit einem der Dienste ConfirmedEventNotification oder UnconfirmedEventNotification verwendet werden.

5.1.5.1 Struktur

Die Struktur der Funktionen des Dienstes AcknowledgeAlarm ist in der folgenden Tabelle dargestellt. In der Tabelle verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

Tabelle 25: Struktur der Funktionen des Dienstes AcknowledgeAlarm

Parametername	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
Acknowledging Process Identifier	M	M(=)		
Event Object Identifier	M	M(=)		
Event State Acknowledged	M	M(=)		
Time Stamp	M	M(=)		
Acknowledgment Source	M	M(=)		
Time Of Acknowledgment	M	M(=)		
Result(+)			S	S(=)
Result(-)			S	S(=)
Error Type			M	M(=)

5.1.5.2 Vorgehensweise des Dienstes

Nach Überprüfen der Gültigkeit der Anfrage versucht der Antwortende, das entsprechende Objekt zu lokalisieren. Ist das Objekt vorhanden und der Parameter „Time Stamp“ stimmt mit der aktuellen Zeit für das zu bestätigende Ereignis überein, wird das Bit der Eigenschaft Acked_Transitions des Objektes, das dem Wert des Parameters „Event State Acknowledged“ entspricht, bestätigt, indem der Wert des Bit um „1“ verändert und Result(+) zurückgegeben wird. Anderfalls wird Result(-) zurückgegeben.

Bei erfolgreicher Bestätigung und Rückgabe von Result(+) wird eine Ereignisbenachrichtigung gesendet, deren Parameter „Notify Type“ ACK_NOTIFICATION entspricht. Die Bestätigungsbenachrichtigung (wenn erforderlich) verwendet denselben Dienst und dieselben Empfänger wie die ursprüngliche Meldung.

5.1.6 ConfirmedCOVNotification-Dienst

Der Dienst ConfirmedCOVNotification wird für die Benachrichtigung von Abonnenten über Änderungen an Eigenschaften eines bestimmten Objektes verwendet. Abonnements für COV-Benachrichtigungen werden mit einem der Dienste SubscribeCOV oder SubscribeCOVProperty erstellt.

5.1.6.1 Struktur

Die Struktur der Funktionen des Dienstes ConfirmedCOVNotification ist in der folgenden Tabelle dargestellt. In der Tabelle verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

Tabelle 26: Struktur der Funktionen des Dienstes ConfirmedCOVNotification

Parametername	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
Subscriber Process Identifier	M	M(=)		
Initiating Device Identifier	M	M(=)		
Monitored Object Identifier	M	M(=)		
Time Remaining	M	M(=)		
List of Values	M	M(=)		
Result(+)			S	S(=)
Result(-)			S	S(=)
Error Type			M	M(=)

5.1.6.2 Vorgehensweise des Dienstes

Nach Überprüfen der Gültigkeit der Anfrage führt der Antwortende die lokalen Aktionen aus, die der angegebenen Werteänderung (COV) zugewiesen wurden und gibt Result(+) zurück. Kann die Dienstanfrage nicht ausgeführt werden, wird Result(-) mit der entsprechenden Fehlerangabe zurückgegeben.

5.1.7 UnconfirmedCOVNotification-Dienst

Der Dienst UnconfirmedCOVNotification wird verwendet, um Abonnenten über Änderungen der Eigenschaften eines bestimmten Objektes zu informieren oder Objekteigenschaften allgemeinen Interesses (z. B. Daten über die Außenluft) gleichzeitig ohne Abonnement an mehrere Geräte zu versenden. Abonnements für COV-Benachrichtigungen werden mit dem Dienst SubscribeCOV erstellt. Die Regel für die Bestimmung, wann dieser Dienst Benachrichtigungen ohne Abonnements versendet, ist lokal unterschiedlich und kann aufgrund von Werteänderungen, regelmäßigen Aktualisierungen oder anderen Kriterien erfolgen.

5.1.7.1 Struktur

Die Struktur der Funktionen des Dienstes UnconfirmedCOVNotification ist in der folgenden Tabelle dargestellt. In der Tabelle verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

Tabelle 27: Struktur der Funktionen des Dienstes UnconfirmedCOVNotification

Parametername	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
Subscriber Process Identifier	M	M(=)		
Initiating Device Identifier	M	M(=)		
Monitored Object Identifier	M	M(=)		
Time Remaining	M	M(=)		
List of Values	M	M(=)		

5.1.7.2 Vorgehensweise des Dienstes

Da dieser Dienst keine Bestätigung erfordert, wird keine Antwort erwartet. Aktionen als Antwort auf Benachrichtigungen dieses Dienstes sind lokal unterschiedlich.

5.1.8 ConfirmedEventNotification-Dienst

Der Dienst ConfirmedEventNotification wird von einem Benachrichtigungsserver verwendet, um ein entferntes Gerät über das Auftreten eines Ereignisses zu informieren, für das der Benachrichtigungsserver eine Bestätigung des Erhalts der Benachrichtigung erfordert. Dies bedeutet, dass das Gerät den Erhalt der Benachrichtigung bestätigt. Es bedeutet nicht, dass ein Anwender informiert wurde. Für die Angabe, ob ein Anwender eine Benachrichtigung, die eine Bestätigung erfordert, erhalten hat, wird der separate Dienst AcknowledgeAlarm verwendet. Müssen mehrere Empfänger benachrichtigt werden, ist für jeden Empfänger eine separate Instanz dieses Dienstes erforderlich. Ist keine Empfangsbestätigung erforderlich, kann UnconfirmedEventNotification verwendet werden.

5.1.8.1 Struktur

Die Struktur der Funktionen des Dienstes ist in der folgenden Tabelle dargestellt. Die verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

Tabelle 28: Struktur der Funktionen des Dienstes ConfirmedEventNotification

Parametername	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
Process Identifier	M	M(=)		
Initiating Device Identifier	M	M(=)		
Event Object Identifier	M	M(=)		
Time Stamp	M	M(=)		
Notification-Class	M	M(=)		
Priority	M	M(=)		
Event Type	M	M(=)		
Message Text	U	U(=)		
Notify Type	M	M(=)		
AckRequired	C	C(=)		
From State	C	C(=)		
To State	M	M(=)		
Event Values	C	C(=)		
Result(+)			S	S(=)
Result(-)			S	S(=)
Error Type			M	M

5.1.8.2 Vorgehensweise des Dienstes

Nach Überprüfen der Gültigkeit der Anfrage führt der Antwortende die lokalen Aktionen aus, die dem angegebenen Ereignis zugewiesen wurden und gibt Result(+) zurück. Kann die Dienstanfrage nicht ausgeführt werden, wird Result(-) mit der entsprechenden Fehlerangabe zurückgegeben.

5.1.9 UnconfirmedEventNotification-Dienst

Der Dienst UnconfirmedEventNotification wird von einem Benachrichtigungsserver verwendet, um ein entferntes Gerät über das Auftreten eines Ereignisses zu informieren. Aufgabe des Dienstes ist die Benachrichtigung von Empfängern über das Auftreten eines Ereignisses, wenn keine Bestätigung über den Empfang der Benachrichtigung erforderlich ist. Anwendungen, die eine Empfangsbestätigung der Benachrichtigung durch das entfernte Gerät erfordern, müssen den Dienst ConfirmedEventNotification verwenden. Die Tatsache, dass dieser Dienst keine Bestätigung erfordert, bedeutet nicht, dass er nicht für Alarmbenachrichtigungen geeignet ist. Ereignisse des Typs Alarm können eine Bestätigung durch einen Anwender erfordern, die mit dem Dienst AcknowledgeAlarm übertragen wird. Folglich hat die Verwendung eines nicht zu bestätigenden Dienstes für Alarmbenachrichtigungen keine Auswirkungen auf eine Empfangsbestätigung durch den Anwender. Alle Geräte, die diesen Dienst ausführen, müssen programmierbare Prozess-Identifizierer verwenden, um die Zuweisung des Parameters „Process Identifier“ für Broadcast- und Multicast-Benachrichtigungen auf Installationsbasis zu ermöglichen.

5.1.9.1 Struktur

Die Struktur der Funktionen des Dienstes UnconfirmedEventNotification ist in der folgenden Tabelle dargestellt. In der Tabelle verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

Tabelle 29: Struktur der Funktionen des Dienstes UnconfirmedEventNotification

Parametername	Req	Ind
Argument	M	M(=)
Process Identifier	M	M(=)
Initiating Device Identifier	M	M(=)
Event Object Identifier	M	M(=)
Time Stamp	M	M(=)
Notification-Class	M	M(=)
Priority	M	M(=)
Event Type	M	M(=)
Message Text	U	U(=)
Notify Type	M	M(=)
AckRequired	C	C(=)
From State	C	C(=)
To State	M	M(=)
Event Values	C	C(=)

5.1.9.2 Vorgehensweise des Dienstes

Da dieser Dienst keine Bestätigung erfordert, wird keine Antwort erwartet. Aktionen als Antwort auf Benachrichtigungen dieses Dienstes sind lokal unterschiedlich.

5.1.10 GetAlarmSummary-Dienst

Der Dienst GetAlarmSummary wird als Zusammenfassung der „Active Alarms“ verwendet. Im aktiven Alarmzustand sind BACnet-Standardobjekte, deren Eigenschaft Event_State einen anderen Wert als NORMAL und deren Eigenschaft Notify_Type den Wert ALARM aufweisen. Der Dienst GetEnrollmentSummary bietet vielfältige Möglichkeiten mit verschiedenen Filtern.

5.1.10.1 Struktur

Die Struktur der Funktionen des Dienstes GetAlarmSummary ist in der folgenden Tabelle dargestellt. In der Tabelle verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

Tabelle 30: Struktur der Funktionen des Dienstes GetAlarmSummary

Parametername	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
Result(+)			S	S(=)
List of Alarm Summaries			M	M(=)
Object Identifier			M	M(=)
Alarm State			M	M(=)
Acknowledged Transitions			M	M(=)
Result(-)			S	S(=)
Error Type			M	M(=)

5.1.10.2 Vorgehensweise des Dienstes

Nach Überprüfen der Gültigkeit der Anfrage durchsucht der Antwortende alle Objekte, die Ereignisse auslösen und deren Eigenschaft Event_State einen anderen Wert als NORMAL sowie deren Eigenschaft Notify_Type den Wert ALARM haben. Es wird eine positive Antwort generiert, die Alarmzusammenfassungen für bei dieser Suche gefundene Objekte enthält. Werden keine Objekte gefunden, die diesen Kriterien entsprechen, wird eine Liste mit der Länge Null zurückgegeben.

5.1.11 GetEnrollmentSummary-Dienst

Der Dienst GetEnrollmentSummary wird für eine Zusammenfassung von Objekten verwendet, die Ereignisse auslösen. Für die Definition der Suchkriterien stehen verschiedene Filter zur Verfügung. Mit diesem Dienst können Zusammenfassungen von Objekten mit beliebigem Event-Type erstellt werden. Der Dienst beinhaltet eine Obermenge der Funktionen des Dienstes GetAlarmSummary.

5.1.11.1 Struktur

Die Struktur der Funktionen des Dienstes GetEnrollmentSummary ist in der folgenden Tabelle dargestellt. In der Tabelle verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

Tabelle 31: Struktur der Funktionen des Dienstes GetEnrollmentSummary

Parametername	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
Acknowledgment Filter	M	M(=)		
Enrollment Filter	U	U(=)		
Event State Filter	U	U(=)		
Event Type Filter	U	U(=)		
Priority Filter	U	U(=)		
Notification-Class Filter	U	U(=)		
Result(+)			S	S(=)
List of Enrollment Summaries			M	M(=)
Object Identifier			M	M(=)
Event Type			M	M(=)
Event State			M	M(=)
Priority			M	M(=)
Notification-Class			U	U(=)
Result(-)			S	S(=)
Error Type			M	M(=)

5.1.11.2 Vorgehensweise des Dienstes

Nach Überprüfen der Gültigkeit der Anfrage sucht der Antwortende nach allen Objekten, die Ereignisse auslösen und den in der Anfrage angegebenen Suchkriterien entsprechen. Die Suchkriterien sind logische Verknüpfungen aller angegebenen Filter sowie der Standardwerte der Filter, die nicht in der Anfrage enthalten sind. Es wird eine positive Antwort mit EnrollmentSummaries der gefundenen Objekte zurückgegeben. Werden keine Objekte gefunden, die diesen Kriterien entsprechen, wird eine Liste mit der Länge Null zurückgegeben.

5.1.12 GetEventInformation-Dienst

Der Dienst GetEventInformation wird für eine Zusammenfassung aller „Active Event States“ verwendet. Einen aktiven Ereignisstatus haben Objekte, die Ereignisse auslösen und deren

- Eigenschaft Event_State einen anderen Wert als NORMAL aufweist oder
- die über die Eigenschaft Acked_Transitions verfügen, bei der mindestens eines der Bits TO-OFFNORMAL, TO-FAULT oder TO-NORMAL auf „FALSE“ gesetzt ist.

Dieser Dienst ist in alle Geräte zu implementieren, die Ereignisbenachrichtigungen generieren.

5.1.12.1 Struktur

Die Struktur der Funktionen des Dienstes GetEventInformation ist in der folgenden Tabelle dargestellt. In der Tabelle verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

Tabelle 32: Struktur der Funktionen des Dienstes GetEventInformation

Parametername	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
Last Received Object Identifier	U	U(=)		
Result(+)				
List of Event Summaries			M	M(=)
Object Identifier			M	M(=)
Event State			M	M(=)
Acknowledged Transitions			M	M(=)
Event Time Stamps			M	M(=)
Notify_Type			M	M(=)
Event Enable			M	M(=)
Event Priorities			M	M(=)
More Events			M	M(=)
Result(-)			S	S(=)
Error Type			M	M(=)

5.1.12.2 Vorgehensweise des Dienstes

Nach Überprüfen der Gültigkeit der Anfrage sucht der Antwortende nach allen Objekten, die Ereignisse auslösen und folgende Bedingungen erfüllen, beginnend mit Objekt nach dem durch den Parameter „Last Received Object Identifier“ angegebenen Objekt, falls vorhanden (je nach interner Reihenfolge der Objekte im antwortenden Gerät):

- Objekte, deren Eigenschaft Event_State einen anderen Wert als NORMAL hat, oder
- Objekte, in deren Eigenschaft Acked_Transitions mindestens eines der Bits TO-OFFNORMAL, TO-FAULT oder TO-NORMAL auf „FALSE“ gesetzt ist.

Es wird eine positive Antwort generiert, die Ereigniszusammenfassungen für Objekte enthält, die bei dieser Suche gefunden wurden. Werden keine Objekte gefunden, die diesen Kriterien entsprechen, wird eine Liste mit der Länge Null zurückgegeben. Es werden so viele Objekte zurückgegeben, wie die APDU zulässt. Entsprechen mehr Objekte den Kriterien, als die APDU zulässt, wird der Parameter „More Events“ auf „TRUE“ gesetzt, andernfalls auf „FALSE“.

5.1.13 LifeSafetyOperation-Dienst

Der Dienst LifeSafetyOperation wird in Brandmelde-, Personenschutz- und Sicherheitssystemen verwendet und bietet eine Methode für die Übergabe bestimmter Anweisungen eines Anwenders, um Folgendes zu erreichen:

- Stummschaltung optischer oder akustischer Benachrichtigungsgeräte
- Zurücksetzen verknüpfter Benachrichtigungsgeräte
- Wiederherstellen stummgeschalteter optischer oder akustischer Benachrichtigungsgeräte

5.1.13.1 Struktur

Die Struktur der Funktionen des Dienstes LifeSafetyOperation ist in der folgenden Tabelle dargestellt. In der Tabelle verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

Tabelle 33: Struktur der Funktionen des Dienstes LifeSafetyOperation

Parametername	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
Requesting Process Identifier	M	M(=)		
Requesting Source	M	M(=)		
Request	M	M(=)		
Object Identifier	U	U(=)		
Result(+)			S	S(=)
Result(-)			S	S(=)
Error Type			M	M(=)

5.1.13.2 Vorgehensweise des Dienstes

Der Antwortende überprüft zuerst die Gültigkeit des Parameters „Object Identifier“ und gibt Result(-) mit der entsprechenden Fehlerklasse und dem entsprechenden Fehlercode zurück, falls der Parameter Request ungültig oder der Parameter „Object Identifier“ vorhanden ist und ein Objekt angibt, das entweder nicht bekannt ist oder keine gültige Anfrage für dieses Objekt darstellt.

Ist der Parameter „Object Identifier“ nicht vorhanden, versucht der Antwortende, alle entsprechenden Objekte des Gerätes basierend auf dem Parameter Request zu handhaben. Ist der Parameter „Object Identifier“ vorhanden, versucht der Antwortende, das in diesem Parameter angegebene Objekt stummzuschalten oder zurückzusetzen, basierend auf der Anfrage des Parameters Request. In beiden Fällen gibt der Antwortende Result(+) zurück.

5.1.14 SubscribeCOV-Dienst

Mit dem Dienst SubscribeCOV abonniert ein COV-Client Änderungsbenachrichtigungen für die Eigenschaften eines bestimmten Objektes. Einige BACnet-Standardobjekte unterstützen COV-Reporting optional. Unterstützt ein Standardobjekt COV-Reporting, lösen Werteänderungen bestimmter Eigenschaften des Objektes (in einigen Fällen basierend auf programmierbaren Schwellenwerten) das Senden von COV-Benachrichtigungen an einen oder mehrere Clients mit Abonnement aus. In der Regel werden COV-Benachrichtigungen an Überwachungsprogramme in BACnet-Client-Geräten, an Anwender oder an Protokollgeräte gesendet. Proprietäre Objekte unterstützen COV-Reporting auf Wunsch des Entwicklers. Standardobjekte, die COV optional unterstützen, sowie die zu verwendenden COV-Algorithmen sind in der Tabelle in Kapitel „Melden bei Wertänderungen (Change Of Value / State Reporting)“ aufgeführt.

Das Abonnement erstellt eine Verbindung zwischen Erkennung der Werteänderung und der Reporting-Methode des COV-Servers sowie einen Prozess im COV-Client-Gerät. Änderungsbenachrichtigungen werden vom COV-Server-Gerät ausgegeben, wenn nach Erstellung des Abonnements Änderungen auftreten. Der COV-Server verwendet die Dienste ConfirmedCOVNotification und UnconfirmedCOVNotification, um die Änderungsbenachrichtigung zu übermitteln. Der Versand zu bestätigender bzw. nicht zu bestätigender Nachrichten wird im Abonnement angegeben.

5.1.14.1 Struktur

Die Struktur der Funktionen des Dienstes SubscribeCOV ist in der folgenden Tabelle dargestellt. In der Tabelle verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

Tabelle 34: Struktur der Funktionen des Dienstes SubscribeCOV

Parametername	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
Subscriber Process Identifier	M	M(=)		
Monitored Object Identifier	M	M(=)		
Issue Confirmed Notifications	U	U(=)		
Lifetime	U	U(=)		
Result(+)			S	S(=)
Result(-)			S	S(=)
Error Type			M	M(=)

5.1.14.2 Vorgehensweise des Dienstes

Ist weder Lifetime noch „Issue Confirmed Notifications“ vorhanden, wird die Anfrage storniert. Jeder COV-Kontext, der bereits für die in der PDU enthaltene BACnet-Adresse vorhanden ist, welche die SubscribeCOV-Anfrage überträgt und über denselben „Subscriber Process Identifier“ und „Monitored Object Identifier“ verfügt, wird storniert und es wird Result(-) zurückgegeben. Stornierungen, für die kein entsprechender COV-Kontext vorhanden ist, werden behandelt als ob ein Kontext besteht und es wird Result(+) zurückgegeben.

Ist der Parameter Lifetime vorhanden und hat den Wert „Null“, das Gerät unterstützt jedoch keine automatische Stornierung von Abonnements, wird Result(-) zurückgegeben. Ist der Parameter Lifetime nicht vorhanden, aber der Parameter „Issue Confirmed Notifications“ ist vorhanden, wird für die Lifetime der Wert „Null“ (unbegrenzt) angenommen. Ist der Parameter „Issue Confirmed Notifications“ vorhanden, aber das zu überwachende Objekt unterstützt kein COV-Reporting, wird Result(-) zurückgegeben. Unterstützt das zu überwachende Objekt COV-Reporting, dann findet eine Suche nach einem bestehenden COV-Kontext für die BACnet-Adresse im PDU statt, welche die SubscribeCOV-Anfrage überträgt und über denselben „Subscriber Process Identifier“ und „Monitored Object Identifier“ verfügt. Wird ein bestehender COV-Kontext gefunden, gilt die Anfrage als Neu-Abonnement und wird wie ein neu erstelltes Abonnement behandelt.

Wird kein COV-Kontext gefunden, der mit der Anfrage übereinstimmt, wird ein neuer COV-Kontext mit der BACnet-Adresse der PDU erstellt, welche die SubscribeCOV-Anfrage überträgt und denselben „Subscriber Process Identifier“ und „Monitored Object Identifier“ hat. Kann kein Kontext erstellt werden, wird Result(-) zurückgegeben.

Bei Erstellung eines neuen Kontextes oder eines Neu-Abonnements wird der COV-Kontext initialisiert und erhält die im entsprechenden Parameter angegebene Lifetime (falls vorhanden) bzw. eine Lifetime von Null, falls der Parameter nicht vorhanden ist. Das Abonnement wird nach Ablauf dieser Zeit in Sekunden automatisch storniert, außer es wird ein Neu-Abonnement empfangen. Eine Lifetime von Null gibt an, dass das Abonnement unbegrenzt gültig ist und keine automatische Stornierung erfolgt. In allen Fällen wird Result(+) zurückgegeben. Nach erfolgreichem Abschluss eines Abonnements oder eines Neu-Abonnements muss möglichst bald wie im Parameter „Issue Confirmed Notifications“ angegeben eine ConfirmedCOVNotification oder UnconfirmedCOVNotification ausgegeben werden.

5.1.15 SubscribeCOVProperty-Dienst

Mit dem Dienst SubscribeCOVProperty abonniert ein COV-Client Änderungsbenachrichtigungen für die Eigenschaften eines bestimmten Objektes. Alle BACnet-Standardobjekte unterstützen optional COV-Reporting. Wird COV-Reporting verwendet, lösen Werteänderungen abonniert Eigenschaften des Objektes (in einigen Fällen basierend auf programmierbaren Schwellenwerten) das Senden von COV-Benachrichtigungen an einen oder mehrere Clients mit Abonnement aus. In der Regel werden COV-Benachrichtigungen an Überwachungsprogramme in BACnet-Client-Geräten, an Anwender oder an Protokollgeräte gesendet.

Das Abonnement erstellt eine Verbindung zwischen der Erkennung einer Werteänderung und deren Reporting-Methode des COV-Servers und einen Prozess im COV-Client-Gerät. Änderungsbenachrichtigungen werden vom COV-Servergerät ausgegeben, wenn nach Erstellung des Abonnements Änderungen auftreten. Der COV-Server verwendet die Dienste ConfirmedCOVNotification und UnconfirmedCOVNotification, um die Änderungsbenachrichtigung zu übermitteln. Der Versand zu bestätigender bzw. nicht zu bestätigender Nachrichten wird im Abonnement angegeben. Proprietäre und Standardobjekte unterstützen COV-Reporting einer beliebigen Eigenschaft auf Wunsch des Entwicklers.

Der Dienst SubscribeCOVProperty unterscheidet sich vom Dienst SubscribeCOV darin, dass er die Überwachung anderer als der in Tabelle in Kapitel „Melden bei Wertänderungen (Change Of Value / State Reporting)“ aufgeführten Eigenschaften ermöglicht.

5.1.15.1 Struktur

Die Struktur der Funktionen des Dienstes SubscribeCOVProperty ist in der folgenden Tabelle dargestellt. In der Tabelle verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

Tabelle 35: Struktur der Funktionen des Dienstes SubscribeCOVProperty

Parametername	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
Subscriber Process Identifier	M	M(=)		
Monitored Object Identifier	M	M(=)		
Issue Confirmed Notifications	U	U(=)		
Lifetime	U	U(=)		
Monitored Property Identifier	M	M(=)		
COV Increment	U	U(=)		
Result(+)			S	S(=)
Result(-)			S	S(=)
Error Type			M	M(=)

5.1.15.2 Vorgehensweise des Dienstes

Ist weder Lifetime noch „Issue Confirmed Notifications“ vorhanden, wird die Anfrage storniert. Jeder COV-Kontext, der bereits für die in der PDU enthaltene BACnet-Adresse vorhanden ist, welche die SubscribeCOVProperty-Anfrage überträgt und über denselben „Subscriber Process Identifier“, „Monitored Object Identifier“ und „Monitored Property Identifier“ verfügt, wird storniert und es wird Result(-) zurückgegeben. Stornierungen, für die kein entsprechender COV-Kontext vorhanden ist, werden behandelt als ob ein Kontext besteht und es wird Result(+) zurückgegeben. Wird ein bestehender COV-Kontext gefunden, wird dieser aus der Eigenschaft Active_COV_Subscriptions des Device-Objektes entfernt.

Ist der Parameter „Issue Confirmed Notifications“ vorhanden, aber die zu überwachende Eigenschaft unterstützt kein COV-Reporting, wird Result(-) zurückgegeben. Unterstützt die zu überwachende Eigenschaft COV-Reporting, dann findet eine Suche nach einem bestehenden COV-Kontext für die BACnet-Adresse im PDU statt, welche die SubscribeCOVProperty-Anfrage überträgt und über denselben „Subscriber Process Identifier“, „Monitored Object Identifier“ und „Monitored Property Identifier“ verfügt. Wird ein bestehender COV-Kontext gefunden, gilt die Anfrage als Neu-Abonnement und wird wie ein neu erstelltes Abonnement behandelt.

Wird kein COV-Kontext gefunden, der mit der Anfrage übereinstimmt, wird ein neuer COV-Kontext mit der BACnet-Adresse der PDU erstellt, welche die SubscribeCOVProperty-Anfrage überträgt und denselben „Subscriber Process Identifier“, „Monitored Object Identifier“ und „Monitored Property Identifier“ besitzt. Der neue Kontext wird der Eigenschaft Active_COV_Subscriptions des Device-Objektes hinzugefügt. Kann kein Kontext erstellt werden, wird Result(-) zurückgegeben.

Bei Erstellung eines neuen Kontextes oder eines Neu-Abonnements wird der COV-Kontext initialisiert und erhält die im entsprechenden Parameter angegebene Lifetime. Das Abonnement wird nach Ablauf dieser Zeit in Sekunden automatisch storniert, außer es wird ein Neu-Abonnement empfangen. Nach erfolgreichem Abschluss eines Abonnements oder eines Neu-Abonnements wird Result(+) zurückgegeben und es muss möglichst bald wie im Parameter „Issue Confirmed Notifications“ angegeben eine ConfirmedCOVNotification oder UnconfirmedCOVNotification ausgegeben werden.

5.2 File-Access-Dienste

Dieser Abschnitt definiert Dienste für den Zugriff und die Bearbeitung von Dateien in BACnet-Geräten. Dateien bezeichnen im Netzwerk sichtbare Oktette beliebiger Länge und Bedeutung. Dies ist lediglich ein abstraktes Konzept und bedeutet nicht, dass Platten-, Band- oder anderen Speichermethoden in Servergeräten verwendet werden. Die entsprechenden Dienste können für den Zugriff auf vom Hersteller definierte als auch auf im BACnet-Protokollstandard definierte Dateien verwendet werden.

Jede Datei, die über File-Access-Dienste erreichbar ist, muss über ein entsprechendes File-Objekt im BACnet-Gerät verfügen. Mit dem File-Objekt wird die Datei namentlich identifiziert. Weiterhin ermöglicht das File-Objekt Zugriff auf Header-Informationen wie Gesamtgröße, Erstellungsdatum oder Typ der Datei. File-Access-Dienste können Dateien auf zwei Arten abbilden: als kontinuierlichen Oktett-Datenstrom oder als zusammenhängende Sequenz nummerierter Records.

File-Access-Dienste ermöglichen Atomic-Lese- und Schreibaktionen. In diesem Kontext bedeutet „Atomic“, dass während der Ausführung einer Lese- oder Schreibaktion keine andere Atomic-Lese- oder Schreibaktion für diese Datei möglich ist. Die Synchronisierung dieser Dienste mit internen Vorgängen des BACnet-Gerätes ist eine lokale Bedingung und in dieser Dokumentation nicht definiert.

5.2.1 AtomicReadFile-Dienst

Der AtomicReadFile-Dienst wird für das Öffnen, Lesen und Schließen des Inhaltes der angegebenen Datei verwendet. Der Zugriff auf die Datei kann record- oder oktettbasiert erfolgen.

5.2.1.1 Struktur

Die Struktur der Funktionen des Dienstes AtomicReadFile ist in der folgenden Tabelle dargestellt. In der Tabelle verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

Tabelle 36: Struktur der Funktionen des Dienstes AtomicReadFile

Parametername	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
File Identifier	M	M(=)		
Stream Access	S	S(=)		
File Start Position	M	M(=)		
Requested Octet Count	M	M(=)		
Record Access	S	M(=)		
File Start Record	M	M(=)		
Requested Record Count	M	M(=)		
Result(+)			S	S(=)
End Of File			M	M(=)
Stream Access			S	S(=)
File Start Position			M	M(=)
File Data			C	C(=)
Record Access			S	S(=)
File Start Record			M	M(=)
Returned Record Count			M	M(=)
File Record Data			C	C(=)
Result(-)			S	S(=)
Error Type			M	S(=)

5.2.1.2 Vorgehensweise des Dienstes

Der Antwortende überprüft zuerst die Gültigkeit des Parameters „File Identifier“ und gibt Result(-) mit entsprechender Fehlerklasse und Fehlercode zurück, falls das File-Objekt ungültig ist, derzeit von einem anderen AtomicReadFile- oder AtomicWriteFile-Dienst verwendet wird oder aus einem anderen Grund derzeit kein Zugriff möglich ist. Ist einer der Parameter „File Start Position“ oder „File Start Record“ geringer als „0“ oder überschreitet die tatsächliche Dateigröße, wird Result(-) mit dem entsprechenden Fehler zurückgegeben. Andernfalls liest der Antwortende die in „Requested Octet Count“ angegebene Oktettanzahl oder die in „Requested Record Count“ angegebenen Records aus. Ist die Anzahl der vorhandenen Oktette oder Einträge geringer als die angefragte Anzahl, gibt die Länge der zurückgegebenen „File Data“ oder der „Returned Record Count“ die tatsächlich ausgelesene Anzahl an. Enthält die Antwort mindestens ein Oktett bzw. einen Record der Datei, ist der Parameter „End Of File“ „TRUE“, andernfalls „FALSE“.

5.2.2 AtomicWriteFile-Dienst

Der AtomicWriteFile-Dienst wird von einem Client-BACnet-User für das Öffnen, Schreiben und Schließen eines OCTET STRING in die angegebenen Position oder einer Liste von OCTET STRINGS in eine bestimmte Gruppe von Einträgen einer Datei verwendet. Der Zugriff auf die Datei kann eintrag- oder oktettbasiert erfolgen.

5.2.2.1 Struktur

Die Struktur der Funktionen des Dienstes AtomicWriteFile ist in der folgenden Tabelle dargestellt. In der Tabelle verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

Tabelle 37: Struktur der Funktionen des Dienstes AtomicWriteFile

Parametername	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
File Identifier	M	M(=)		
Stream Access	S	S(=)		
File Start Position	M	M(=)		
File Data	M	M(=)		
Record Access	S	S(=)		
File Start Record	M	M(=)		
Record Count	M	M(=)		
File Record Data	M	M(=)		
Result(+)			S	S(=)
Stream Access			S	S(=)
File Start Position			M	M(=)
Record Access			S	S(=)
File Start Record			M	M(=)
Result(-)			S	S(=)
Error Type			M	M(=)

5.2.2.2 Vorgehensweise des Dienstes

Der Antwortende überprüft zuerst die Gültigkeit des Parameters „File Identifier“ und gibt Result(-) mit der entsprechenden Fehlerklasse und dem entsprechenden Fehlercode zurück, falls das File-Objekt ungültig ist, derzeit von einem anderen AtomicReadFile- oder AtomicWriteFile-Dienst verwendet wird oder aus einem anderen Grund derzeit kein Zugriff möglich ist. Überschreitet einer der Parameter „File Start Position“ oder „File Start Record“ die tatsächliche Dateigröße, wird die Datei auf die angegebene Größe erweitert, der Inhalt dazwischenliegender Oktette oder Einträge ist lokal unterschiedlich. Hat einer dieser Parameter den Sonderwert „-1“, beginnt der Schreibvorgang am derzeitigen Ende der Datei. In diesem Fall schreibt der Antwortende die in „Octet Count“ angegebene Oktettanzahl oder die in „Record Count“ angegebenen Records in die Datei. Schlägt der Schreibvorgang fehl, wird Result(-) mit entsprechender Fehlerklasse und Fehlercode zurückgegeben. Ist der gesamte Schreibvorgang erfolgreich, wird Result(+) zurückgegeben. „File Start Position“ oder „File Start Record“ geben die tatsächliche Position an, an der Daten geschrieben wurden.

5.3 Object-Access-Dienste

In diesem Abschnitt werden neun Anwendungsdienste beschrieben, die für den Zugriff und die Bearbeitung der Eigenschaften von BACnet-Objekten verwendet werden. Ein BACnet-Objekt ist jedes Objekt, auf dessen Eigenschaften über dieses Protokoll zugegriffen werden kann, unabhängig der Funktion des Gerätes, in dem sich das Objekt befindet. Die Dienste können für den Zugriff auf vom Hersteller definierte und auf in dieser Dokumentation definierte Objekte verwendet werden.

5.3.1 AddListElement-Dienste

Der Dienst AddListElement wird von BACnet-Clients verwendet, um zu einer Objekteigenschaft, die eine Liste ist, eines oder mehrere Elemente hinzuzufügen.

5.3.1.1 Struktur

Die Struktur der Funktionen des Dienstes AddListElement ist in der folgenden Tabelle dargestellt. In der Tabelle verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

Tabelle 38: Struktur der Funktionen des Dienstes AddListElement

Parametername	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
Object Identifier	M	M(=)		
Property Identifier	M	M(=)		
Property Array Index	C	C(=)		
List of Elements	M	M(=)		
Result(+)			S	S(=)
Result(-)			S	S(=)
Error Type			M	M(=)
First Failed Element Number			M	M(=)

5.3.1.2 Vorgehensweise des Dienstes

Nach Überprüfen der Gültigkeit der Anfrage versucht der Antwortende, das im Parameter „Object Identifier“ angegebene Objekt zu bearbeiten. Ist das Objekt vorhanden und besitzt es die im Parameter „Property Identifier“ angegebene Eigenschaft, wird versucht, der Eigenschaft alle im Parameter „List of Elements“ angegebenen Elemente hinzuzufügen. Ist dieser Versuch erfolgreich, wird Result(+) zurückgegeben. Sind ein oder mehrere Elemente bereits in der Liste enthalten, werden diese ignoriert, d. h. nicht der Liste hinzugefügt. Ignorieren eines bereits vorhandenen Elementes ist kein Fehler des Dienstes.

Ist das angegebene Objekt oder die angegebene Eigenschaft nicht vorhanden oder die Eigenschaft ist keine Liste, schlägt der Dienst fehl und es wird Result(-) zurückgegeben. Kann eines oder mehrere Elemente der Liste nicht hinzugefügt werden, die nicht in der Liste enthalten sind, wird Result(-) zurückgegeben und der Liste werden keine Elemente hinzugefügt.

Dieser Dienst fügt der Liste entweder alle angegebenen Elemente hinzu, die noch nicht in der Liste enthalten sind, oder es wird kein Element hinzugefügt.

5.3.2 RemoveListElement-Dienst

Der Dienst RemoveListElement wird verwendet, um eines oder mehrere Elemente aus einer Objekteigenschaft zu löschen, die eine Liste ist. Ist eines der Elemente selbst eine Liste, wird das gesamte Element entfernt. Dieser Dienst kann nicht mit verschachtelten Listen verwendet werden.

5.3.2.1 Struktur

Die Struktur der Funktionen des Dienstes RemoveListElement ist in der folgenden Tabelle dargestellt. In der Tabelle verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

Tabelle 39: Struktur der Funktionen des Dienstes RemoveListElement

Parametername	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
Object Identifier	M	M(=)		
Property Identifier	M	M(=)		
Property Array Index	C	C(=)		
List of Elements	M	M(=)		
Result(+)			S	S(=)
Result(-)			S	S(=)
Error Type			M	M(=)
First Failed Element Number			M	M(=)

5.3.2.2 Vorgehensweise des Dienstes

Nach Überprüfen der Gültigkeit der Anfrage versucht der Antwortende, das im Parameter „Object Identifier“ angegebene Objekt zu bearbeiten. Ist das Objekt vorhanden und besitzt es die im Parameter „Property Identifier“ angegebene Eigenschaft, wird versucht, alle im Parameter „List of Elements“ angegebenen Elemente aus der Eigenschaft zu löschen. Sind ein oder mehrere der Elemente nicht vorhanden oder können aufgrund fehlender Berechtigungen nicht entfernt werden, wird keines der Elemente entfernt und es wird Result(-) zurückgegeben.

5.3.3 ReadProperty-Dienst

Der Dienst ReadProperty wird von einem BACnet-Client verwendet, um den Wert einer Eigenschaft eines BACnet-Objektes abzufragen. Mit diesem Dienst kann jede Eigenschaft jedes Objektes ausgelesen werden, unabhängig davon, ob das BACnet-Objekt definiert ist oder nicht.

5.3.3.1 Struktur

Die Struktur der Funktionen des Dienstes ReadProperty ist in folgenden Tabelle dargestellt. In der Tabelle verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

Tabelle 40: Struktur der Funktionen des Dienstes ReadProperty

Parametername	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
Object Identifier	M	M(=)		
Property Identifier	M	M(=)		
Property Array Index	U	U(=)		
Result(+)			S	S(=)
Object Identifier			M	M(=)
Property Identifier			M	M(=)
Property Array Index			U	U(=)
Property Value			M	M(=)
Result(-)			S	S(=)
Error Type			M	M(=)

5.3.3.2 Vorgehensweise des Dienstes

Nach Überprüfen der Gültigkeit der Anfrage versucht der Antwortende, auf die angegebene Eigenschaft des angegebenen Objektes zuzugreifen. Bei erfolgreichem Zugriff wird Result(+) mit dem Wert der Eigenschaft zurückgegeben. Schlägt der Zugriff fehl, wird Result(-) mit Angabe des Fehlergrundes zurückgegeben.

Enthält das Objekt des Parameters „Object Identifier“ den Wert „Device Object“ und die Instanz des Parameters „Object Identifier“ den Wert „4194303“, behandelt der Antwortende den „Object Identifier“, als ob er ordnungsgemäß mit dem lokalen Device-Objekt übereinstimmt. Dies ermöglicht die Bestimmung der Geräteinstanz eines Gerätes, das keine I-Am-Meldung generiert.

5.3.4 ReadPropertyConditional-Dienst

Der Dienst ReadPropertyConditional wird verwendet, um „Object Identifier“ und Werte von Null oder mehr angegebenen Eigenschaften aller BACnet-Objekte abzufragen, die eine Liste von Auswahlkriterien oder Bedingungen erfüllen. Die Bedingungen sind Boolean-Ausdrücke im Format (Property.Relational_Operator.Constant). Die Liste kann mit Bedingungen mit logisch OR oder AND beginnen, um festzustellen, ob ein bestimmtes Objekt zu den auszulesenden Objekten hinzuzufügen ist. Eine weitere Möglichkeit ist, dass alle Objekte ausgewählt und deren angegebene Eigenschaften ausgelesen werden.

Bei dem Dienst ReadProperty sind im Gegensatz zum Dienst ReadPropertyConditional die Objekte, deren Eigenschaften ausgelesen werden sollen, ausdrücklich vom BACnet-Client anzugeben. Bei dem Dienst ReadPropertyConditional hingegen werden die Objekte durch den Antwortenden mithilfe der übermittelten Auswahlkriterien bestimmt.

5.3.4.1 Struktur

Die Struktur der Funktionen des Dienstes ReadPropertyConditional ist in der folgenden Tabelle dargestellt. In der Tabelle verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

Tabelle 41: Struktur der Funktionen des Dienstes ReadPropertyConditional

Parametername	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
Object Selection Criteria	M	M(=)		
List of Property References	U	U(=)		
Result(+)			S	S(=)
List of Read Access Results			M	M(=)
Result(-)			S	S(=)
Error Type			M	M(=)

5.3.4.2 Vorgehensweise des Dienstes

Nach Überprüfen der Gültigkeit der Anfrage durchsucht der Antwortende seine Objektdatenbank nach Objekten, die den angegebenen Auswahlkriterien entsprechen. Für jedes gefundene Objekt wird ein „Read Access Result“ gemäß der „List of Property References“ generiert. Auch wenn die Anfrage nicht gemeinsam durchgeführt werden muss, hat der Anfragende dennoch sicherzustellen, dass alle Abfragen schnellstmöglich durchgeführt und nur durch Prozesse mit höherer Priorität unterbrochen werden. Die Ausführung der Anfrage wird fortgesetzt, bis ein Zugriffsversuch auf alle angegebenen Eigenschaften der ausgewählten Objekte erfolgt ist. Werden keine Objekte gefunden, die den angegebenen Kriterien entsprechen, wird Result(-) und eine

„List of Results“ mit der Länge Null zurückgegeben. Werden ein oder mehrere Objekte gefunden, die den Auswahlkriterien entsprechen, wird Result(+) mit den „Read Access Results“ aller gefundenen Übereinstimmungen zurückgegeben.

5.3.4.3 Parameter des Dienstes ReadPropertyConditional

Folgende Parameter können in den Anweisungen des Dienstes ReadPropertyConditional enthalten sein:

5.3.4.3.1 Object Selection Criteria

Der Parameter „Object Selection Criteria“ ist in der folgenden Tabelle dargestellt. In der Tabelle verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

Tabelle 42: Struktur des Parameters „Object Selection Criteria“

Parametername	Req Ind	Rsp Cnf	Datentyp
Selection Logic	M	M(=)	Enumerated
List of Selection Criteria	C	C(=)	
Property Identifier	M	M(=)	BACnetPropertyIdentifier
Property Array Index	C	C(=)	Unsigned
Relation Specifier	M	M(=)	Enumerated
Comparison Value	M	M(=)	ANY

5.3.4.3.2 Read Access Result

Der Parameter „Read Access Result“ ist in der folgenden Tabelle dargestellt. In der Tabelle verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

Tabelle 43: Struktur des Parameters „Read Access Result“

Parametername	Rsp	Cnf	Datentyp
Object Identifier	M	M(=)	BACnetObjectIdentifier
List of Results	U	U(=)	
Property Identifier	M	M(=)	BACnetPropertyIdentifier
Property Array Index	U	U(=)	Unsigned
Property Value	S	S(=)	ANY
Property Access Error	S	S(=)	Error

5.3.5 ReadPropertyMultiple-Dienst

Der Dienst ReadPropertyMultiple wird verwendet, um die Werte einer oder mehrerer Eigenschaften eines oder mehrerer BACnet-Objekte anzufragen. Mit diesem Dienst ist Lesezugriff auf beliebige Eigenschaften beliebiger Objekte möglich, unabhängig davon, ob das Objekt in BACnet definiert ist oder nicht. Der Anwender kann eine einzelne Eigenschaft eines einzelnen Objektes, eine Liste von Eigenschaften eines Objektes oder mehrere Eigenschaften mehrerer Objekte auslesen. Eine „Read Access Specification“ mit dem „Property Identifier“ ALL kann für die Ermittlung der in ein Objekt implementierten Eigenschaften sowie derer Werte verwendet werden.

5.3.5.1 Struktur

Die Struktur der Funktionen des Dienstes ReadPropertyMultiple ist in folgenden Tabelle dargestellt. In der Tabelle verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

Tabelle 44: Struktur der Funktionen des Dienstes ReadPropertyMultiple

Parametername	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
List of Read Access Specifications	M	M(=)		
Result(+)			S	S(=)
List of Read Access Results			M	M(=)
Result(-)			S	S(=)
Error Type			M	M(=)

5.3.5.2 Vorgehensweise des Dienstes

Nach Überprüfen der Gültigkeit der Anfrage versucht der Antwortende, auf die angegebenen Eigenschaften der angegebenen Objekte zuzugreifen und generiert eine „List of Read Access Results“ in der in der Anfrage angegebenen Reihenfolge. Enthält der Abschnitt „List of Property References“ des Parameters „List of Read Access Specifications“ den „Property Identifier“ ALL, REQUIRED oder OPTIONAL, wird die „List of Read Access Results“ generiert, als ob auf jede zurückgegebene Eigenschaft explizit verwiesen wäre (siehe Kapitel „Gesamtliste der BACnet-Dienst-Parameter“, Parameter „List of Property References“). Auch wenn die Anfrage nicht gemeinsam durchgeführt werden muss, hat der Anfragende dennoch sicherzustellen, dass alle Abfragen schnellstmöglich durchgeführt und nur durch Prozesse mit höherer Priorität unterbrochen werden. Die Ausführung der Anfrage wird fortgesetzt bis ein Zugriffsversuch auf alle angegebenen Eigenschaften erfolgt ist. Wird keines der angegebenen Objekte gefunden oder ist der Zugriff auf keine der angegebenen Eigenschaften der angegebenen Objekte möglich, wird entweder Result(-) oder Result(+) mit Fehlercodes für alle Eigenschaften zurückgegeben. Ist der Zugriff auf mindestens eine der angegebenen Eigenschaften der angegebenen Objekte möglich, wird Result(+) mit allen ausgelesenen Werten und Fehlercodes für alle Eigenschaften, auf die kein Zugriff möglich war, zurückgegeben.

Enthält das Objekt des Abschnittes „Object Identifier“ des Parameters „Read Access Specification“ den Wert „Device Object“ und des Parameters „Object Identifier“ den Wert „4194303“, behandelt der Antwortende den „Object Identifier“, als ob er ordnungsgemäß mit dem lokalen Device-Objekt übereinstimmt. Dies ermöglicht die Bestimmung der Geräteinstanz eines Gerätes, das keine I-Am-Meldung generiert.

5.3.5.3 Parameter des Dienstes ReadPropertyMultiple

Folgende Parameter können in den Anweisungen des Dienstes ReadPropertyMultiple enthalten sein:

5.3.5.3.1 Read Access Specification

Der Parameter „Read Access Specification“ ist in der folgenden Tabelle dargestellt. In der Tabelle verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

Tabelle 45: Struktur des Parameters Read Access Specification

Parametername	ReqInd	RspCnf	Datentyp
Object Identifier	M	M(=)	BACnetObjectIdentifier
List of Property References	M	M(=)	List of BACnetPropertyReference

5.3.5.3.2 Read Access Result

Der Parameter „Read Access Result“ ist in der folgenden Tabelle dargestellt. In der Tabelle verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

Tabelle 46: Struktur des Parameters „Read Access Result“

Parametername	Rsp	Cnf	Datentyp
Object Identifier	M	M(=)	BACnetObjectIdentifier
List of Results	M	M(=)	BACnetPropertyIdentifier Unsigned Beliebig Error
Property Identifier	M	M(=)	
Property Array Index	U	U(=)	
Property Value	S	S(=)	
Property Access Error	S	S(=)	Error

5.3.6 ReadRange-Dienst

Der Dienst ReadRange wird von einem BACnet-Client verwendet, um einen bestimmten Bereich von Datenelementen auszulesen, die eine Untergruppe der in einer bestimmten Objekteigenschaft verfügbaren Daten darstellen. Dieser Dienst kann für Eigenschaften aus einer beliebigen Liste oder einem Feld aus Listen verwendet werden.

5.3.6.1 Struktur

Die Struktur der Funktionen des Dienstes ReadRange ist in der folgenden Tabelle dargestellt. In der Tabelle verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

Tabelle 47: Struktur der Funktionen des Dienstes ReadRange

Parametername	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
Object Identifier	M	M(=)		
Property Identifier	M	M(=)		
Property Array Index	C	C(=)		
Range	U	U(=)		
Result(+)			S	S(=)
Object Identifier			M	M(=)
Property Identifier			M	M(=)
Property Array Index			C	C(=)
Result Flags			M	M(=)
Item Count			M	M(=)
Item Data			M	M(=)
First Sequence Number			C	C(=)
Result(-)			S	S(=)
Error Type			M	M(=)

5.3.6.1.1 Range

Tabelle 48: Struktur des Parameters Range

Parametername	Req	Ind	Datentyp
By Position	S	S(=)	
Reference Index	M	M(=)	Unsigned
Count	M	M(=)	INTEGER
By Sequence Number	S	S(=)	
Reference Sequence Number	M	M(=)	Unsigned32
Count	M	M(=)	INTEGER
By Time	S	S(=)	
Reference Time	M	M(=)	BACnetDateTime
Count	M	M(=)	INTEGER

5.3.6.2 Vorgehensweise des Dienstes

Der Antwortende überprüft zuerst die Gültigkeit der Parameter „Object Identifier“, „Property Identifier“ und „Property Array Index“ und gibt

Result(-) mit der entsprechenden Fehlerklasse und dem entsprechenden Fehlercode zurück, falls das Objekt oder die Eigenschaft ungültig ist, die referenzierten Daten keine Liste oder kein Feld sind oder derzeit aus einem anderen Grund kein Zugriff möglich ist.

Ist der Parameter Range nicht vorhanden, liest der Antwortende alle verfügbaren Elemente der Liste oder des Feldes aus und versucht diese zurückzugeben.

Ist der Parameter Range vorhanden und gibt den Parameter „By Position“ an, liest der Antwortende alle angegebenen Elemente aus und versucht, diese zurückzugeben. Die angegebenen Elemente umfassen das im Index als „Reference Index“ angegebene Element bis „Count -1“, falls Count positiv ist, oder bis „-1 - Count“, falls Count negativ ist. Das erste Element einer Liste hat den Index 1.

Ist der Parameter Range vorhanden und gibt die Parameter „By Time“ an, liest der Antwortende alle angegebenen Elemente aus und versucht, diese zurückzugeben. Sind Parameter „By Time“ vorhanden und die Werte der Eigenschaft verfügen nicht über einen Zeitstempel, wird ein Fehler zurückgegeben. Ist Count positiv, umfassen die angegebenen Einträge den ersten Eintrag mit einem aktuelleren Zeitstempel als „Reference Time“ bis zum Element „Count -1“. Ist Count negativ, umfassen die angegebenen Einträge den ersten Eintrag mit einem aktuelleren Zeitstempel als Reference Time bis zum Element „-1 - Count“. Die Sequenznummer des ersten zurückgegebenen Elementes muss in der Antwort enthalten sein. Die Elemente werden in chronologischer Reihenfolge zurückgegeben.

Ist der Parameter Range vorhanden und gibt die Parameter „By Sequence Number“ an, liest der Antwortende alle angegebenen Elemente aus und versucht, diese zurückzugeben. Angegebene Elemente sind alle Elemente mit einer Sequenznummer im Bereich „Reference Sequence Number“ bis „Reference Sequence Number plus Count -1“, falls Count positiv ist, oder im Bereich „Reference Sequence Number plus Count +1“ bis „Reference Sequence“, falls Count negativ ist.

Um das Auslassen von Elementen bei verketteten zeitbasierten Lesevorgängen zu vermeiden, wird das erste Element des angegebenen Bereiches mit dem Format „By Time“ des Parameters Range gesucht. Weitere Anfragen zum Auslesen der verbleibenden Elemente des angegebenen Bereiches erfolgen im Format „By Sequence Number“ des Parameters Range. Grund hierfür ist, dass Listen mit Zeitstempel, die nach Eingangszeit sortiert sind, unter Umständen über Einträge mit aus der Reihe fallenden Zeitstempeln verfügen, da die lokale Zeit des Gerätes zurückgestellt wurde. Werden Objekte ausgelesen, die den angefragten Parametern entsprechen, aber nicht in der Antwort zurückgegeben werden können, ist das Flag MORE_ITEMS des Parameters „Result Flags“ auf „TRUE“ gesetzt, andernfalls auf „FALSE“. Verbleibende Elemente können durch Angabe entsprechender Parameter mit weiteren Anfragen abgefragt werden.

Die Antwort enthält die Anzahl der ausgelesenen und zurückgegebenen Elemente im Parameter „Item Count“. Die eigentlichen Elemente werden im Parameter „Item Data“ zurückgegeben. Enthält die Antwort den ersten Positionsindex und es wurde eine „By Position“-Anfrage durchgeführt, oder die älteste Sequenznummer und es wurde eine „By Sequence Number“- oder „By Time“-Anfrage durchgeführt, ist das Flag FIRST_ITEM des Parameters „Result Flags“ auf „TRUE“ gesetzt, andernfalls auf „FALSE“.

Enthält die Antwort den letzten Positionsindex und es wurde eine „By Position“-Anfrage durchgeführt oder die aktuellste Sequenznummer und es wurde eine „By Sequence“-Number- oder „By Time“-Anfrage durchgeführt, ist das Flag LAST_ITEM des Parameters „Result Flags“ auf „TRUE“ gesetzt, andernfalls auf „FALSE“.

Sind keine Elemente in der Liste enthalten, die den Kriterien des Parameters Range entsprechen, wird Result(+) mit einem Item Count von „0“ und ohne dem Parameter „First Sequence Number“ zurückgegeben.

5.3.7 WriteProperty-Dienst

Der Dienst WriteProperty wird verwendet, um den Wert einer einzelnen angegebenen Eigenschaft eines BACnet-Objektes zu bearbeiten. Mit diesem Dienst ist Schreibzugriff auf beliebige Eigenschaften beliebiger Objekte möglich, unabhängig davon, ob das Objekt in BACnet definiert ist oder nicht. Einige Entwickler beschränken unter Umständen den Schreibzugriff auf bestimmte Eigenschaften bestimmter Objekte. In diesem Fall wird beim Versuch, das Objekt zu bearbeiten, ein Fehler mit der Fehlerklasse PROPERTY und dem Fehlercode WRITE_ACCESS_DENIED zurückgegeben.

5.3.7.1 Struktur

Die Struktur der Funktionen des Dienstes WriteProperty ist in der folgenden Tabelle dargestellt. In der Tabelle verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

Tabelle 49: Struktur der Funktionen des Dienstes WriteProperty

Parametername	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
Object Identifier	M	M(=)		
Property Identifier	M	M(=)		
Property Array Index	U	U(=)		
Property Value	M	M(=)		
Priority	C	C(=)		
Result(+)			S	S(=)
Result(-)			S	S(=)
Error Type			M	M(=)

5.3.7.2 Vorgehensweise des Dienstes

Nach Überprüfen der Gültigkeit der Anfrage versucht der Antwortende, die angegebene Eigenschaft des angegebenen Objektes auf den im Parameter „Property Value“ angegebenen Wert zu ändern. Ist dieser Versuch erfolgreich, wird Result(+) zurückgegeben. Schlägt der Schreibzugriff fehl, wird Result(-) mit Angabe des Fehlergrundes zurückgegeben. Der bedingte Parameter Priority wird wie in Kapitel „Gesamtliste der BACnet-Dienst-Parameter“, Parameter „Priority“ beschrieben verwendet.

5.3.8 WritePropertyMultiple-Dienst

Der Dienst WritePropertyMultiple wird verwendet, um die Werte einer oder mehrerer Eigenschaften eines BACnet-Objektes zu bearbeiten. Mit diesem Dienst ist im Grunde Schreibzugriff auf beliebige Eigenschaften beliebiger Objekte möglich, unabhängig davon, ob das Objekt in BACnet definiert ist oder nicht.

Eigenschaften werden mit dem Dienst WritePropertyMultiple in der Reihenfolge bearbeitet, die im Parameter „List of Write Access Specifications“ angegeben ist. Die Ausführung des Dienstes wird fortgesetzt, bis alle angegebenen Eigenschaften geschrieben wurden oder der Vorgang auf eine Eigenschaft trifft, die nicht wie angegeben bearbeiten werden kann.

5.3.8.1 Struktur

Die Struktur der Funktionen des Dienstes WritePropertyMultiple ist in der folgenden Tabelle dargestellt. In der Tabelle verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

Tabelle 50: Struktur der Funktionen des Dienstes WritePropertyMultiple

Parametername	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
List of Write Access Specifications	M	M(=)		
Result(+)			S	S(=)
Result(-)			S	S(=)
Error Type			M	M(=)
First Failed Write Attempt			M	M(=)

5.3.8.2 Vorgehensweise des Dienstes

Für jede in der „List of Write Access Specifications“ enthaltene „Write Access Specification“ wird der Wert jeder angegebenen Eigenschaft mit dem Wert aus „Write Access Specification“ ersetzt und es wird Result(+) zurückgegeben, um anzuzeigen, dass die Dienstanfrage vollständig ausgeführt wurde. Der bedingte Parameter Priority wird wie in Kapitel „Gesamtliste der BACnet-Dienst-Parameter“, Parameter „Priority“ beschrieben verwendet.

Wird bei der Bearbeitung der angegebenen Eigenschaften in der Reihenfolge, welche die „List of Write Access Specifications“ vorgibt, eine Eigenschaft angetroffen, die nicht bearbeitet werden kann, gibt der Antwortende Result(-) mit Angabe des Fehlergrundes zurück. Mit diesem Dienst werden entweder alle angegebenen Eigenschaften oder nur die Eigenschaften bis exklusiv der im Parameter „First Failed Write Attempt“ angegebenen Eigenschaft erfolgreich bearbeitet.

5.3.8.3 Parameter des Dienstes WritePropertyMultiple

Der Parameter „Write Access Specification“ ist in der folgenden Tabelle dargestellt. In der Tabelle verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

Tabelle 51: Struktur des Parameters „Write Access Specification“

Parametername	Req Ind	Rsp Cnf	Datentyp
Object Identifier	M	M(=)	BACnetObjectIdentifier
List of Properties	M	M(=)	BACnetPropertyIdentifier Unsigned Beliebig Unsigned (1..16)
Property Identifier	M	M(=)	
Property Array Index	U	U(=)	
Property Value	M	M(=)	
Priority	C	C(=)	

5.4 Remote-Device-Management-Dienste

5.4.1 DeviceCommunicationControl-Dienst

Der Dienst DeviceCommunicationControl wird verwendet, um entfernte Geräte im Kommunikationsnetzwerk oder -Internetwork für eine bestimmte Zeit anzuweisen, keine APDUs zu initiieren und optional keine APDUs zu beantworten (mit Ausnahme von DeviceCommunicationControl oder, wenn unterstützt, ReinitializeDevice). Der Dienst DeviceCommunicationControl wird hauptsächlich von einem Anwender für Diagnosezwecke verwendet. Die Dauer kann auf unendlich gesetzt werden, d. h. die Kommunikation muss mit dem Dienst DeviceCommunicationControl oder ReinitializeDevice neu gestartet werden.

5.4.1.1 Struktur

Die Struktur der Funktionen des Dienstes DeviceCommunicationControl ist in der folgenden Tabelle dargestellt. In der Tabelle verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

Tabelle 52: Struktur der Funktionen des Dienstes DeviceCommunicationControl

Parametername	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
Time Duration	U	U(=)		
Enable/Disable	M	M(=)		
Password	U	U(=)		
Result(+)			S	S(=)
Result(-)			S	S(=)
Error Type			M	M(=)

5.4.1.2 Vorgehensweise des Dienstes

Nach Überprüfen der Gültigkeit der Anfrage und des Kennwortes gibt der Antwortende Result(+) zurück, und antwortet auf keine weiteren Meldungen außer DeviceCommunicationControl und ReinitializeDevice und beendet das Initiieren von Meldungen, falls der Parameter Enable/Disable auf DISABLE gesetzt ist. Die Kommunikation ist deaktiviert bis entweder „Time Duration“ abgelaufen ist oder eine gültige DeviceCommunicationControl- (mit Enable/Disable = ENABLE) oder eine gültige ReinitializeDevice-Meldung eingegangen ist. Ist das Kennwort ungültig oder erforderlich und nicht vorhanden, wird die APDU ignoriert und Result(-) zurückgegeben.

5.4.2 ReinitializeDevice-Dienst

Der Dienst ReinitializeDevice wird verwendet, um ein entferntes Gerät zu einem Neustart (Kaltstart) oder zum Zurücksetzen auf einen vorbestimmten Zustand (Warmstart) aufzufordern oder den Sicherungs- oder Wiederherstellungsvorgang zu steuern. Das Zurücksetzen oder Neustarten von Geräten wird hauptsächlich von einem Anwender für Diagnosezwecke durchgeführt. Die Verwendung dieses Dienstes bei Sicherungs- oder Wiederherstellungsvorgängen wird in der Regel vom Gerät initiiert, das die Sicherung oder Wiederherstellung steuert. Aus Gründen der Vertraulichkeit erfordert die Ausführung dieses Dienstes unter Umständen die Übergabe eines Kennwortes.

5.4.2.1 Struktur

Die Struktur der Funktionen des Dienstes ReinitializeDevice ist in der folgenden Tabelle dargestellt. In der Tabelle verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

Tabelle 53: Struktur der Funktionen des Dienstes ReinitializeDevice

Parametername	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
Reinitialized State of Device	M	M(=)		
Password	U	U(=)		
Result(+)			S	S(=)
Result(-)			S	S(=)
Error Type			M	M(=)

5.4.2.2 Vorgehensweise des Dienstes

Nach Überprüfen der Gültigkeit der Anfrage und des Kennwortes hat der antwortende BACnet-Client Vorrang vor allen anderen ausstehenden Anfragen und gibt Result(+) zurück. Ist die Anfrage WARMSTART oder COLDSTART, führt der antwortende BACnet-Client den entsprechenden Vorgang unverzüglich durch, anschließend wird das Gerät wie in der Anfrage angegeben neu initialisiert. Ist die Dienstanfrage WARMSTART und ist das Gerät aufgrund interner Vorgänge nicht bereit, wird Result(-) zurückgegeben.

Ist das Kennwort ungültig oder erforderlich und nicht vorhanden, wird Result(-) zurückgegeben.

5.4.3 TimeSynchronization-Dienst

Der Dienst TimeSynchronization wird verwendet, um die korrekte Zeit an ein entferntes Gerät zu übermitteln. Dieser Dienst kann als Broadcast- oder Multicast an mehrere oder an einen einzelnen Empfänger gerichtet sein. Dank der Benachrichtigungen über die korrekte Zeit können Geräte ihre internen Uhren synchronisieren.

5.4.3.1 Struktur

Die Struktur der Funktionen des Dienstes TimeSynchronization ist in der folgenden Tabelle dargestellt. In der Tabelle verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

Tabelle 54: Struktur der Funktionen des Dienstes TimeSynchronization

Parametername	Req	Ind
Argument	M	M(=)
Time	M	M(=)

5.4.3.2 Vorgehensweise des Dienstes

Da dieser Dienst keine Bestätigung erfordert, wird keine Antwort erwartet. Ein Gerät, das eine TimeSynchronization-Anfrage empfängt, aktualisiert seine lokale Zeit. Diese Änderung wird in den Eigenschaften Local_Time und Local_Date des Gerätes angegeben.

Bei Aufruf dieses Dienstes durch einen Anwender bestehen keine Einschränkungen. Andernfalls wird die Verwendung des Dienstes durch den Wert der Eigenschaft Time_Synchronization_Recipients des Device-Objektes geregelt. Ist die Länge der Liste Time_Synchronization_Recipients Null, darf das Gerät keine automatischen TimeSynchronization-Anfragen senden. Ist die Länge der Liste Time_Synchronization_Recipients „1“ oder größer, darf das Gerät automatisch TimeSynchronization-Anfragen senden, jedoch nur an die in der Liste Time_Synchronization_Recipients aufgeführten Geräte oder Adressen.

5.4.4 UTCTimeSynchronization-Dienst

Der Dienst UTCTimeSynchronization wird von einem anfragenden BACnet-Anwender verwendet, um die korrekte UTC-Zeit an ein oder mehrere entfernte Geräte zu übermitteln. Dieser Dienst kann als Broadcast- oder Multicast-Dienst an mehrere oder an einen einzelnen Empfänger gerichtet sein. Dank der Benachrichtigungen über die korrekte UTC-Zeit können Geräte ihre internen Uhren synchronisieren.

5.4.4.1 Struktur

Die Struktur der Funktionen des Dienstes UTCTimeSynchronization ist in der folgenden Tabelle dargestellt. In der Tabelle verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

Tabelle 55: Struktur der Funktionen des Dienstes UTCTimeSynchronization

Parametername	Req	Ind
Argument	M	M(=)
Time	M	M(=)

5.4.4.2 Vorgehensweise des Dienstes

Da dieser Dienst keine Bestätigung erfordert, wird keine Antwort erwartet. Ein Gerät, das eine UTCTimeSynchronization-Meldung erhält, aktualisiert unter Berücksichtigung der Eigenschaft Daylight_Savings_Status des Device-Objektes seine lokale Zeit und sein lokales Datum durch Abzug des Wertes der Eigenschaft UTC_Offset des Device-Objektes vom Parameter Time. Diese Änderung wird in den Eigenschaften Local_Time und Local_Date des Gerätes angegeben.

Bei Aufruf dieses Dienstes durch einen Anwender bestehen keine Einschränkungen. Andernfalls wird der Aufruf des Dienstes durch den Wert der Eigenschaft Time_Synchronization_Recipients des Device-Objektes geregelt. Ist die Länge der Liste Time_Synchronization_Recipients Null, darf das Gerät keine automatischen TimeSynchronization-Anfragen senden. Ist die Länge der Liste Time_Synchronization_Recipients „1“ oder größer, darf das Gerät automatisch UTCTimeSynchronization-Anfragen senden, jedoch nur an die in der Liste Time_Synchronization_Recipients aufgeführten Geräte oder Adressen.

5.4.5 Dienste Who-Has und I-Have

Der Who-Has-Dienst wird für die Bestimmung der Device-Objekt-Identifizierung anderer BACnet-Geräte verwendet, deren lokale Datenbank ein Objekt mit einem bestimmten Object_Name oder Object_Identifier enthält. Der I-Have-Dienst wird für die Antwort auf Who-Has-Anfragen verwendet, um das Vorhandensein eines bestimmten Object_Name oder Object_Identifier anzuzeigen. Eine I-Have-Meldung kann jederzeit ausgegeben werden und erfordert keine vorherige Who-Has-Anfrage. Die Dienste Who-Has und I-Have erfordern keine Bestätigung.

5.4.5.1 Struktur des Dienstes Who-Has

Die Struktur der Funktionen des Dienstes Who-Has ist in der folgenden Tabelle dargestellt. In der Tabelle verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

Tabelle 56: Struktur der Funktionen des Dienstes Who-Has

Parametername	Req	Ind
Argument	M	M(=)
Device Instance Range Low Limit	U	U(=)
Device Instance Range High Limit	U	U(=)
Object Identifier	S	S(=)
Object Name	S	S(=)

5.4.5.2 Vorgehensweise des Dienstes Who-Has

Der Sendende übermittelt die Who-Has-Anfrage in der Regel über eine Broadcast-Adresse. Falls die Parameter „Device Instance Range Low Limit“ und „Device Instance Range High Limit“ vorhanden sind, dürfen nur BACnet-Geräte antworten, deren Device-Object_Identifier-Instanznummer im Bereich von „Device Instance Range Low Limit <= Object_Identifier Instance Number“ <= „Device Instance Range High Limit“ liegt. Ist der Parameter „Object Name“ vorhanden, antworten nur BACnet-Geräte mit einer I-Have-Meldung, die ein Objekt enthalten, dessen Eigenschaft Object_Name mit dem Parameter „Object Name“ übereinstimmt. Ist der Parameter „Object Identifier“ vorhanden, antworten nur BACnet-User mit einer I-Have-Meldung, die ein Objekt enthalten, dessen Eigenschaft Object_Identifier mit dem Parameter „Object Identifier“ übereinstimmt.

5.4.5.3 Struktur des Dienstes I-Have

Die Struktur der Funktionen des Dienstes I-Have ist in der folgenden Tabelle dargestellt. In der Tabelle verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

Tabelle 57: Struktur der Funktionen des Dienstes I-Have

Parametername	Req	Ind
Argument	M	M(=)
Device Identifier	M	M(=)
Object Identifier	M	M(=)
Object Name	M	M(=)

5.4.5.4 Vorgehensweise des Dienstes I-Have

Der sendende BACnet-User übermittelt die I-Have-Anfrage in der Regel über eine Broadcast-Adresse. Diese Broadcasts können je nach Anwendungseinstellung über das lokale Netzwerk, ein entferntes Netzwerk oder global über alle Netzwerke erfolgen. Bei Übertragung der I-Have-Meldung als Antwort auf eine Who-Has-Anfrage muss die Übertragung so erfolgen, dass das BACnet-Gerät, welches die Anfrage gesendet hat, auch die Antwort erhält. Die Anfrage erfordert keine Bestätigung, es ist also keine weitere Aktion erforderlich. Ein BACnet-Gerät kann jederzeit eine I-Have-Meldung senden.

5.4.6 Dienste Who-Is und I-Am

Der Dienst Who-Is wird zur Bestimmung des „Object Identifier“, der Netzwerkadresse oder beider Angaben anderer BACnet-Geräte im selben Internetwork verwendet. Der Dienst Who-Is erfordert keine Bestätigung. Der Dienst kann für die Bestimmung des „Object Identifier“ und der Netzwerkadresse aller Geräte im Netzwerk oder die Bestimmung der Netzwerkadresse eines bestimmten Gerätes verwendet werden, dessen Object Identifier jedoch nicht dessen Adresse bekannt ist. Der Dienst I-Am erfordert ebenfalls keine Bestätigung. Der Dienst I-Am wird für die Beantwortung von Who-Is-Anfragen verwendet. Eine I-Am-Meldung kann jederzeit gesendet werden. Es ist keine vorherige Who-Is-Anfrage erforderlich. Ein Gerät kann beispielsweise beim Anschalten eine I-Am-Broadcast-Meldung versenden. Die Netzwerkadresse wird entweder von MAC-Adressen abgeleitet, die der I-Am-Dienstanfrage zugewiesen sind, falls sich das Gerät, das die Anfrage sendet, im lokalen Netzwerk befindet, oder vom NPCI, falls sich das Gerät in einem entfernten Netzwerk befindet.

5.4.6.1 Struktur des Dienstes Who-Is

Die Struktur der Funktionen des Dienstes Who-Is ist in der folgenden Tabelle dargestellt. In der Tabelle verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

Tabelle 58: Struktur der Funktionen des Dienstes Who-Is

Parametername	Req	Ind
Argument	M	M(=)
Device Instance Range Low Limit	U	U(=)
Device Instance Range High Limit	U	U(=)

5.4.6.2 Vorgehensweise des Dienstes Who-Is

Das sendende BACnet-Gerät übermittelt die Who-Is-Anfrage in der Regel über eine Broadcast-Adresse. Falls die Parameter „Device Instance Range Low Limit“ und „Device Instance Range High Limit“ nicht vorhanden sind, müssen alle empfangenden BACnet-Geräte ihren Object_Identifier in einzelnen Antworten mit dem Dienst I-Am zurückgeben. Sind die Parameter „Device Instance Range Low Limit“ und „Device Instance Range High Limit“ vorhanden, dürfen nur BACnet-Geräte ihren Object_Identifier mit dem I-Am-Dienst zurückgeben, deren Device-Object_Identifier-Instanznummer im Bereich von „Device Instance Range Low Limit <= Object_Identifier Instance Number <= Device Instance Range High Limit“ liegt.

Verfügt das empfangende BACnet-Gerät über die Eigenschaft Slave_Proxy_Enable und außerdem ist Slave_Proxy_Enable des empfangenden Anschlusses „TRUE“, antwortet das BACnet-Gerät mit einer nicht zu bestätigenden I-Am-Meldung für jedes Slave-Gerät im MS/TP-

Netzwerk, das in der Eigenschaft Slave_Address_Binding vorhanden ist und den Parametern für „Device Range“ entspricht. Die I-Am-Meldungen werden so

generiert, als ob das Slave-Gerät den Dienst gestartet hat. Erfolgt die I-Am-Anfrage im MS/TP-Netzwerk, in dem sich der Slave befindet, ist die im Paket enthaltene MAC-Adresse die des Slave-Gerätes. Erfolgt die I-Am-Anfrage in einem anderen Netzwerk als der Slave, enthält die Netzwerkschicht SLEN und SNET mit der Slave-MAC-Adresse, als ob das geroutete Paket ursprünglich vom Slave-Gerät erstellt würde.

5.4.6.3 Struktur des Dienstes I-Am

Die Struktur der Funktionen des Dienstes I-Am ist in der folgenden Tabelle dargestellt. In der Tabelle verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

Tabelle 59: Struktur der Funktionen des Dienstes I-Am

Parametername	Req	Ind
Argument	M	M(=)
I-Am Device Identifier	M	M(=)
Max APDU Length Accepted	M	M(=)
Segmentation Supported	M	M(=)
Vendor Identifier	M	M(=)

5.4.6.4 Vorgehensweise des Dienstes I-Am

Der sendende BACnet-User übermittelt die I-Am-Anfrage in der Regel über eine Broadcast-Adresse. Diese Broadcasts können je nach Anwendungseinstellung über das lokale Netzwerk, ein entferntes Netzwerk oder global über alle Netzwerke erfolgen. Bei Übertragung der I-Am-Meldung als Antwort auf eine Who-Is-Anfrage muss die Übertragung so erfolgen, dass das BACnet-Gerät, welches die Anfrage gesendet hat, auch die Antwort erhält. Da die Anfrage keine Bestätigung erfordert, ist keine weitere Aktion erforderlich. Ein BACnet-Gerät kann jederzeit eine I-Am-Meldung senden.

6 Fehler-, Ablehnungs- und Abbruchcodes

Alle im BACnet-Protokoll vorkommenden Fehler sind gemäß einer Kategorie, der sogenannten Fehlerklasse (Error Class) spezifiziert. In jeder Fehlerklasse sind die Fehler einzeln mit Fehlercodes klassifiziert.

6.1 Fehlerklasse – DEVICE

Diese Fehlerklasse beschreibt Fehler, welche die Funktion des gesamten BACnet-Gerätes beeinträchtigen. Das Auftreten eines dieser Fehler bedeutet in der Regel, dass die gesamte Dienstanfrage fehlgeschlagen ist.

Tabelle 60: Fehlerklasse - DEVICE

Fehlercode	Bedeutung
DEVICE_BUSY	Eine Dienstanfrage wurde temporär abgelehnt, da das adressierte BACnet-Gerät erwartet, länger als für den üblichen Timeout für Anfragen/Bestätigungen mit der Bearbeitung einer höheren Priorität beschäftigt zu sein.
CONFIGURATION_IN_PROGRESS	Eine Dienstanfrage wurde temporär abgelehnt, da das adressierte BACnet-Gerät gerade konfiguriert wird, entweder lokal am Gerät oder mithilfe weiterer Protokolldienste.
OPERATIONAL_PROBLEM	Eine Dienstanfrage wurde temporär abgelehnt, da das adressierte BACnet-Gerät ein Problem im Betrieb erkannt hat, das die Ausführung des angefragten Dienstes verhindert.
OTHER	Dieser Fehlercode wird bei anderen als den bisher in dieser Fehlerklasse genannten Gründen zurückgegeben.

6.2 Fehlerklasse – OBJECT

Diese Fehlerklasse beschreibt Fehler bei der Identifikation, dem Zugriff und der Bearbeitung von BACnet-Objekten, unabhängig davon, ob diese in BACnet definiert sind oder nicht. Da diese Fehler in der Regel einzelne Objektmerkmale betreffen, bedeuten sie in der Regel nicht, dass die gesamte Dienstanfrage fehlgeschlagen hat.

Tabelle 61: Fehlerklasse - OBJECT

Fehlercode	Bedeutung
DYNAMIC_CREATION_NOT_SUPPORTED	Es wurde versucht, ein Objekt zu erstellen, welches nicht dynamisch erstellt werden kann.
NO_OBJECTS_OF_SPECIFIED_TYPE	Beim Durchsuchen der Objektdatenbank des angegebenen BACnet-Gerätes wurden keine Objekte des in der Dienstanfrage angegebenen Objektes verwendet.
OBJECT_DELETION_NOT_PERMITTED	Es wurde versucht, ein Objekt zu löschen, das nicht gelöscht werden kann oder derzeit gegen Löschen geschützt ist.
OBJECT_IDENTIFIER_ALREADY_EXISTS	Es wurde versucht, ein neues Objekt mit einem bereits verwendeten Object_Identifier zu erstellen.
READ_ACCESS_DENIED	Es wurde versucht, die Eigenschaften eines Objektes auszulesen, das nicht für den Lesezugriff durch BACnet-Protokolldienste definiert ist.
UNKNOWN_OBJECT	Es wurde ein Object_Identifier für ein Objekt angegeben, das nicht in der Objektdatenbank des angegebenen BACnet-Gerätes enthalten ist.
UNSUPPORTED_OBJECT_TYPE	In einem Dienstparameter wurde ein Objekt angegeben, das nicht bekannt ist oder vom angegebenen BACnet-Gerät nicht unterstützt wird.
OTHER	Dieser Fehlercode wird bei anderen als den bisher in dieser Fehlerklasse genannten Gründen zurückgegeben.

6.3 Fehlerklasse – PROPERTY

Diese Fehlerklasse beschreibt Fehler mit Identifikation, Zugriff und Bearbeitung der Eigenschaften von BACnet-Objekten, unabhängig davon, ob diese in BACnet definiert sind oder nicht. Da diese Fehler in der Regel einzelne Eigenschaftsmerkmale betreffen, bedeuten sie in der Regel nicht, dass die gesamte Dienst-anfrage fehlgeschlagen hat.

Tabelle 62: Fehlerklasse - PROPERTY

Fehlercode	Bedeutung
CHARACTER_SET_NOT_SUPPORTED	Es wurde eine Zeichenkette verwendet, die in diesem Zeichensatz nicht unterstützt wird.
DATATYPE_NOT_SUPPORTED	Die Daten haben bzw. enthalten einen Datentyp, der von dieser Instanz der Eigenschaft nicht unterstützt wird.
INCONSISTENT_SELECTION_CRITERION	Es wurde auf eine Eigenschaft verwiesen, deren Datentyp nicht dem im Dienstparameter „Object Selection Criteria“ angegebenen „Comparison Value“ entspricht. Dieser Fehler tritt beispielsweise auf, wenn ein Analogwert mit einer Boolean-Konstante verglichen wird oder umgekehrt.
INVALID_ARRAY_INDEX	Es wurde versucht, mit einem Feldindex außerhalb des für dieses Feld zugelassenen Bereiches auf eine Feldeigenschaft zuzugreifen.
INVALID_DATATYPE	Der Datentyp einer in einem Dienstparameter angegebenen Eigenschaft entspricht nicht dem Datentyp, der durch den Property_Identifier referenzierten Eigenschaft.
NOT_COV_PROPERTY	Die Eigenschaft wird nicht durch eine COV-Benachrichtigung übertragen.
OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED	Es wurde versucht, einen Wert in eine Eigenschaft zu schreiben, für den das Gerät nicht unterstützte optionale Funktionen bereitstellen müsste.
PROPERTY_IS_NOT_AN_ARRAY	Es wurde versucht, auf eine Eigenschaft als Feld zuzugreifen. Die Eigenschaft verfügt jedoch über keinen Feld-Datentypen.
READ_ACCESS_DENIED	Es wurde versucht, eine Eigenschaft auszulesen, die nicht für den Lesezugriff durch BACnet-Protokolldienste definiert ist.
UNKNOWN_PROPERTY	In einem Dienstparameter wurde ein Property_Identifier angegeben, der nicht bekannt ist oder durch die Objekte des angegebenen BACnet-Gerätes nicht unterstützt wird.
VALUE_OUT_OF_RANGE	Es wurde versucht, einen Wert außerhalb des für diese Eigenschaft definierten Wertebereiches in eine Eigenschaft zu schreiben.
WRITE_ACCESS_DENIED	Es wurde versucht, in eine Eigenschaft zu schreiben, die nicht für den Schreibzugriff durch BACnet-Protokolldienste definiert ist.
OTHER	Dieser Fehlercode wird bei anderen als den bisher in dieser Fehlerklasse genannten Gründen zurückgegeben.

6.4 Fehlerklasse – RESOURCES

Diese Fehlerklasse beschreibt Probleme mit den Ressourcen eines BACnet-Gerätes, welche die Ausführung der Anfragen von Protokolldiensten beeinträchtigen.

Tabelle 63: Fehlerklasse - RESOURCES

Fehlercode	Bedeutung
NO_SPACE_FOR_OBJECT	Der Versuch, ein Objekt zu erstellen, ist fehlgeschlagen, da im angegebenen BACnet-Gerät zu wenig dynamischer Speicher vorhanden ist.
NO_SPACE_TO_ADD_LIST_ELEMENT	Der Versuch, einer Liste ein Element hinzuzufügen, ist fehlgeschlagen, da im angegebenen BACnet-Gerät zu wenig dynamischer Speicher vorhanden ist.
NO_SPACE_TO_WRITE_PROPERTY	Der Versuch, eine Eigenschaft zu schreiben, ist fehlgeschlagen, da im angegebenen BACnet-Gerät zu wenig dynamischer Speicher vorhanden ist.
OTHER	Dieser Fehlercode wird bei anderen als den bisher in dieser Fehlerklasse genannten Gründen zurückgegeben.

6.5 Fehlerklasse – SECURITY

Diese Fehlerklasse beschreibt Fehler bei der Ausführung von Sicherheitsdiensten. Es handelt sich um schwere Fehler, die angeben, dass der Antwortende den angegebenen Dienst nicht ausführen kann.

Tabelle 64: Fehlerklasse - SECURITY

Fehlercode	Bedeutung
AUTHENTICATION_FAILED	Die zu authentifizierende Meldung wurde nicht von diesem Dienstanbieter generiert.
CHARACTER_SET_NOT_SUPPORTED	Es wurde eine Zeichenkette angetroffen, die in diesem Zeichensatz nicht unterstützt wird.
INCOMPATIBLE_SECURITY_LEVELS	Die beiden Clients haben eine unterschiedliche Sicherheitsautorisierung.
INVALID_OPERATOR_NAME	Der Operator-Name ist keinem bekannten Anwender zugewiesen.
KEY_GENERATION_ERROR	Der Schlüsselservice konnte keinen Session Key (SK) generieren.
PASSWORD_FAILURE	Operator-Name und Operator-Passwort stimmen nicht überein.
SECURITY_NOT_SUPPORTED	Der entfernte Client unterstützt keine BACnet-Sicherheitsmethoden.
TIMEOUT	Die APDU mit der erwarteten Invoke-ID wurde nicht vor Ablauf der Wartezeit empfangen.
OTHER	Dieser Fehlercode wird bei anderen als den bisher in dieser Fehlerklasse genannten Gründen zurückgegeben.

6.6 Fehlerklasse – SERVICES

Diese Fehlerklasse beschreibt Fehler bei der Ausführung von Anfragen durch Protokolldienste, unabhängig davon, ob diese in BACnet definiert sind oder nicht. Es handelt sich um schwere Fehler, die angeben, dass der Antwortende den angegebenen Dienst nicht ausführen kann.

Tabelle 65: Fehlerklasse - SERVICES

Fehlercode	Bedeutung
CHARACTER_SET_NOT_SUPPORTED	Es wurde eine Zeichenkette angetroffen, die in diesem Zeichensatz nicht unterstützt wird.
COV_SUBSCRIPTION_FAILED	Das COV-Abonnement ist fehlgeschlagen.
DUPLICATE_NAME	Es wurde versucht, eine Eigenschaft Object_Name mit einem Wert zu schreiben, der bereits für einen anderen Object_Name dieses Gerätes verwendet wird.
DUPLICATE_OBJECT_ID	Es wurde versucht, eine Eigenschaft Object_Identifier mit einem Wert zu schreiben, der bereits für einen anderen Object_Identifier dieses Gerätes verwendet wird.
FILE_ACCESS_DENIED	Wird als Antwort auf eine AtomicReadFile- oder AtomicWriteFile-Anfrage beim Zugriff auf eine Datei generiert, die derzeit gesperrt ist oder auf die aus einem anderen Grund kein Zugriff möglich ist.
INCONSISTENT_PARAMETERS	Es besteht ein Konflikt, da sich zwei oder mehr Parameter der Dienstanfrage gegenseitig ausschließen.
INVALID_CONFIGURATION_DATA	Die angegebenen Konfigurationsdaten sind ungültig oder beschädigt.
INVALID_FILE_ACCESS_METHOD	Wird als Antwort auf eine AtomicReadFile- oder AtomicWriteFile-Anfrage generiert, die eine für die angegebene Datei ungültige File-Access-Methode angibt.
INVALID_FILE_START_POSITION	Wird als Antwort auf eine AtomicReadFile- oder AtomicWriteFile-Anfrage generiert, die einen ungültigen Parameter „File Start Position“ oder „File Start Record“ angibt.
INVALID_PARAMETER_DATATYPE	Der Datentyp eines für einen Dienstparameter angegebenen Wertes ist für den Parameter nicht gültig.
INVALID_TIME_STAMP	Der von einer AcknowledgeAlarm-Dienstanfrage übergebene Parameter „Time Stamp“ stimmt nicht mit der Zeit des letzten Auftretens des zu bestätigenden Ereignisses überein.
MISSING_REQUIRED_PARAMETER	Ein für die Ausführung einer Dienstanfrage erforderlicher Parameter wurde nicht angegeben.
OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED	Die Parameter eines Dienstes erfordern, dass das Gerät nicht unterstützte optionale Funktionen bereitstellt.
PROPERTY_IS_NOT_A_LIST	Es wurde versucht, mit dem Dienst AddListElement oder RemoveListElement auf eine Eigenschaft zuzugreifen, die über keinen Listen-Datentypen verfügt.
PROPERTY_IS_NOT_AN_ARRAY	Es wurde versucht, auf eine Eigenschaft als Feld zuzugreifen. Die Eigenschaft verfügt jedoch über keinen Felddatentypen.
SERVICE_REQUEST_DENIED	Es wurde versucht, einen Dienst auszuführen, für den das anfragende BACnet-Gerät autorisiert ist.
OTHER	Dieser Fehlercode wird bei anderen als den bisher in dieser Fehlerklasse genannten Gründen zurückgegeben.

6.7 Fehlerklasse - VT

Diese Fehlerklasse beschreibt Fehler mit der Ausführung von Virtual-Terminal-Diensten.

Tabelle 66: Fehlerklasse - VT

Fehlercode	Bedeutung
UNKNOWN_VT_CLASS	Dieser Fehler gibt an, dass die in einer VT-Open-Anfrage angegebene VT-Class vom Zielgerät nicht erkannt wurde.
UNKNOWN_VT_SESSION	Dieser Fehler gibt an, dass die in einer VT-Data- oder VT-Close-Anfrage angegebene „VT-Session ID“ vom Zielgerät nicht erkannt wurde.
NO_VT_SESSIONS_AVAILABLE	Dieser Fehler gibt an, dass das Zielgerät die VT-Open-Anfrage aufgrund eingeschränkter Ressourcen nicht ausführen konnte.
VT_SESSION_ALREADY_CLOSED	Dieser Fehler zeigt einen Versuch an, eine bereits beendete VT-Session zu schließen.
VT_SESSION_TERMINATION_FAILURE	Dieser Fehler gibt an, dass eine in einer VT-Close-Anfrage angegebene VT-Session aus Implementierungsgründen nicht freigegeben werden konnte.
OTHER	Dieser Fehlercode wird bei anderen als den bisher in dieser Fehlerklasse genannten Gründen zurückgegeben.

6.8 Ablehnungsgründe

Es können nur PDUs abgelehnt werden, die eine Bestätigung erfordern. Mögliche Gründe für das Ablehnen einer PDU sind in dieser Tabelle aufgeführt:

Tabelle 67: Ablehnungsgründe

Fehlercode	Bedeutung
BUFFER_OVERFLOW	Die Kapazität eines Eingangsspeichers wurde überschritten.
INCONSISTENT_PARAMETERS	Wird als Antwort auf eine zu bestätigende APDU generiert, die eine erforderliche Dienstbedingung nicht enthält oder die eine Dienstbedingung enthält, die nicht vorhanden sein sollte. Diese Bedingung könnte auch zu einer „Reject PDU“ mit Ablehnungsgrund INVALID_TAG führen.
INVALID_PARAMETER_DATA_TYPE	Wird als Antwort auf eine zu bestätigende APDU generiert, bei der die Codierung eines oder mehrerer Dienstparameter nicht der Typspezifikation entspricht. Diese Bedingung könnte auch zu einer „Reject PDU“ mit Ablehnungsgrund INVALID_TAG führen.
INVALID_TAG	Beim Parsen der Meldung ist ein ungültiges Tag aufgetreten. Da ein ungültiges Tag die Parser-Logik verwirren könnte, können als Antwort auf die Anfrage mit dem ungültigen Tag zudem folgende Ablehnungsgründe auftreten: INCONSISTENT_PARAMETERS, INVALID_PARAMETER_DATA_TYPE, MISSING_REQUIRED_PARAMETER und TOO_MANY_ARGUMENTS.
MISSING_REQUIRED_PARAMETER	Wird als Antwort auf eine zu bestätigende APDU generiert, bei der mindestens ein erforderliches Dienstargument fehlt. Diese Bedingung könnte auch zu einer Reject PDU mit Ablehnungsgrund INVALID_TAG führen.
PARAMETER_OUT_OF_RANGE	Wird als Antwort auf eine zu bestätigende APDU generiert, die den Parameter übergibt, dessen Wert außerhalb des für diesen Dienst definierten Bereiches liegt.
TOO_MANY_ARGUMENTS	Wird als Antwort auf eine zu bestätigende APDU generiert, in der die Gesamtanzahl der Dienstargumente größer ist als für den Dienst angegeben. Diese Bedingung könnte auch zu einer Reject PDU mit Ablehnungsgrund INVALID_TAG führen.
UNDEFINED_ENUMERATION	Wird als Antwort auf eine zu bestätigende APDU generiert, bei einer oder mehrere Dienstparameter nicht gemäß der Typspezifikation des entsprechenden Parameters decodiert werden.
UNRECOGNIZED_SERVICE	Generiert als Antwort auf eine zu bestätigende APDU, bei der das Feld Service Choice einen unbekannten oder nicht unterstützten Dienst angibt.
OTHER	Generiert als Antwort auf eine zu bestätigende APDU mit einem nicht näher definierten Syntaxfehler.

6.9 Abbruchgründe

Tabelle 68: Abbruchgründe

Fehlercode	Bedeutung
BUFFER_OVERFLOW	Die Kapazität eines Eingangsspeichers wurde überschritten.
INVALID_APDU_IN_THIS_STATE	Wird als Antwort auf eine APDU generiert, die im aktuellen Zustand von „Transaction State Machine“ nicht erwartet wird.
PREEMPTED_BY_HIGHER_PRIORITY_TASK	Die Aktion wird beendet, um einen Prozess mit höherer Priorität durchzuführen.
SEGMENTATION_NOT_SUPPORTED	Wird als Antwort auf eine APDU generiert, deren Segmentation Bit auf „TRUE“ gesetzt ist, während das empfangende Gerät keine Segmentierung unterstützt. Wird auch generiert, wenn ein BACnet-ComplexACK-PDU so groß ist, dass eine Segmentierung erforderlich ist und nicht übertragen werden kann, da das sendende oder empfangende Gerät keine Segmentierung unterstützt.
OTHER	Dieser Abbruchcode wird bei anderen als den bisher genannten Gründen zurückgegeben.

7 Gesamtliste der BACnet-Objekt-Eigenschaften

Die Eigenschaften von BACnet-Objekten werden nachfolgend in alphabetischer Reihenfolge aufgeführt. Für jede Eigenschaft werden der BACnet-Datentyp sowie seine Entsprechung in der IEC-Applikation aufgeführt.

Die Objektliste zeigt die Objekte, welche diese Eigenschaft besitzen. Anschließend wird die Eigenschaft beschrieben.

Eine Eigenschaft kann unterschiedliche Datentypen besitzen, je nachdem, in welchem Objekt sie vorkommt und zu welchem Zweck sie dort verwendet wird. In diesem Fall entnehmen Sie die Datentypen der Objektliste.

Einige Eigenschaften haben je nach Objekt, in dem sie vorkommen, unterschiedliche Bedeutungen. Diese werden in der Beschreibung getrennt voneinander aufgeführt.

7.1 Accepted_Modes

Datentyp	List of BACnetLifeSafetyMode
Datentyp IEC	ARRAY [0..3] of BACnetLifeSafetyMode
In Objekt	Life-Safety-Point, Life-Safety-Zone
Beschreibung	Diese nur lesbare Eigenschaft des Typs „List of BACnetLifeSafetyMode“ gibt alle Werte an, welche die Eigenschaft Mode beim Schreiben mit BACnet-Diensten akzeptiert. Auch wenn ein Mode in dieser Liste aufgeführt ist, kann das Schreiben vom Objekt aufgrund des aktuellen internen Zustandes des Objektes verweigert werden. Der Wert der Eigenschaft Accepted_Modes hängt nicht vom internen Zustand des Objektes ab und ändert sich nicht, wenn sich der interne Zustand ändert. Bei Verweigern eines Schreibvorganges wird Result(-) mit Angabe der Fehlerklasse PROPERTY und des Fehlercodes VALUE_OUT_OF_RANGE zurückgegeben. Interne Berechnungen des Objektes setzen die Eigenschaft Mode unter Umständen auf einen anderen Wert als in der Eigenschaft Accepted_Modes angegeben.

7.2 Ack_Required

Datentyp	BACnetEventTransitionBits
Datentyp IEC	BACnetEventTransitionBits
In Objekt	Notification-Class
Beschreibung	Diese Eigenschaft des Typs BACnetEventTransitionBits übergibt drei Flags, die angeben, ob für generierte TO-OFFNORMAL-, TO-FAULT- und TO-NORMAL-Ereignisse eine Bestätigung erforderlich ist.

7.3 Acked_Transitions

Datentyp	BACnetEventTransitionBits
Datentyp IEC	BACnetEventTransitionBits
In Objekt	Accumulator, Analog-Value, Binary-Value, Life-Safety-Point, Life-Safety-Zone, Multi-State-Value, Pulse-Converter, Trend-Log
Beschreibung	In Objekt Accumulator: Diese Eigenschaft des Typs BACnetEventTransitionBits übergibt drei Flags, die den Erhalt der Bestätigung für TO-OFFNORMAL-, TO-FAULT- und TO-NORMAL-Ereignisse anzeigen. Im Kontext von Accumulator-Objekten werden Übergänge in die Event_States High_Limit und Low_Limit als OFFNORMAL-Ereignisse angesehen. Die Flags werden nach Auftreten des entsprechenden Ereignisses gelöscht und bei folgenden Bedingungen gesetzt: • bei Erhalt der entsprechenden Bestätigung • bei Auftreten des Ereignisses, falls das entsprechende Flag in der Eigenschaft Event_Enable nicht gesetzt ist (d. h. für diesen Zustand werden keine Ereignisbenachrichtigungen erstellt und folglich wird keine Bestätigung erwartet) • bei Auftreten des Ereignisses, falls das entsprechende Flag in der Eigenschaft Event_Enable gesetzt ist und das entsprechende Flag in der Eigenschaft Ack_Required des Notification-Class-Objektes, auf das die Eigenschaft Notification_Class dieses Objektes eindeutig verweist, nicht gesetzt ist (d. h. es wird keine Bestätigung erwartet)

Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt.

In Objekt Analog-Value:

Diese Eigenschaft des Typs BACnetEventTransitionBits übergibt drei Flags, die den Erhalt der Bestätigung für TO-OFFNORMAL-, TO-FAULT- und TO-NORMAL-Ereignisse anzeigen. Im Kontext von Analog-Value-Objekten werden Wechsel in die Event_States High_Limit und Low_Limit als OFFNORMAL-Ereignisse angesehen. Die Flags werden nach Auftreten des entsprechenden Ereignisses gelöscht und bei folgenden Bedingungen gesetzt:

- bei Erhalt der entsprechenden Bestätigung
- bei Auftreten des Ereignisses, falls das entsprechende Flag in der Eigenschaft Event_Enable nicht gesetzt ist (d.h. für diesen Zustand werden keine Ereignisbenachrichtigungen erstellt und folglich wird keine Bestätigung erwartet)
- bei Auftreten des Ereignisses, falls das entsprechende Flag in der Eigenschaft Event_Enable gesetzt ist und das entsprechende Flag in der Eigenschaft Ack_Required des Notification-Class-Objektes, auf das die Eigenschaft Notification_Class dieses Objektes eindeutig verweist, nicht gesetzt ist (d.h. es wird keine Bestätigung erwartet)

Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt.

In Objekt Binary-Value:

Diese Eigenschaft des Typs BACnetEventTransitionBits übergibt drei Flags, die den Erhalt der Bestätigung für TO-OFFNORMAL-, TO-FAULT- und TO-NORMAL-Ereignisse anzeigen. Die Flags werden nach Auftreten des entsprechenden Ereignisses gelöscht und bei folgenden Bedingungen gesetzt:

- bei Erhalt der entsprechenden Bestätigung
- bei Auftreten des Ereignisses, falls das entsprechende Flag in der Eigenschaft Event_Enable nicht gesetzt ist (d. h. für diesen Zustand werden keine Ereignisbenachrichtigungen erstellt und folglich wird keine Bestätigung erwartet)
- bei Auftreten des Ereignisses, falls das entsprechende Flag in der Eigenschaft Event_Enable gesetzt ist und das entsprechende Flag in der Eigenschaft Ack_Required des Notification-Class-Objektes, auf das die Eigenschaft Notification_Class dieses Objektes eindeutig verweist, nicht gesetzt ist (d. h. es wird keine Bestätigung erwartet)

Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt.

In Objekt Life-Safety-Point:

Diese optionale Eigenschaft des Typs BACnetEventTransitionBits übergibt drei Flags, die den Erhalt der Bestätigung für TO-OFFNORMAL-, TO-FAULT- und TO-NORMAL-Ereignisse anzeigen. Die Flags werden nach Auftreten des entsprechenden Ereignisses gelöscht und bei folgenden Bedingungen gesetzt:

- bei Erhalt der entsprechenden Bestätigung
- bei Auftreten des Ereignisses, falls das entsprechende Flag in der Eigenschaft Event_Enable nicht gesetzt ist (d. h. für diesen Zustand werden keine Ereignisbenachrichtigungen erstellt und folglich wird keine Bestätigung erwartet)
- bei Auftreten des Ereignisses, falls das entsprechende Flag in der Eigenschaft Event_Enable gesetzt ist und das entsprechende Flag in der Eigenschaft Ack_Required des Notification-Class-Objektes, auf das die Eigenschaft Notification_Class dieses Objektes eindeutig verweist, nicht gesetzt ist (d. h. es wird keine Bestätigung erwartet)

Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt.

In Objekt Life-Safety-Zone:

Diese optionale Eigenschaft des Typs

BACnetEventTransitionBits übergibt drei Flags, die den Erhalt der Bestätigung für TO-OFFNORMAL-, TO-FAULT- und TO-NORMAL-Ereignisse anzeigen. Die Flags werden nach Auftreten des entsprechenden Ereignisses gelöscht und bei folgenden Bedingungen gesetzt:

- bei Erhalt der entsprechenden Bestätigung
- bei Auftreten des Ereignisses, falls das entsprechende Flag in der Eigenschaft Event_Enable nicht gesetzt ist (d. h. für diesen Zustand werden keine Ereignisbenachrichtigungen erstellt und folglich wird keine Bestätigung erwartet)
- bei Auftreten des Ereignisses, falls das entsprechende Flag in der Eigenschaft Event_Enable gesetzt ist und das entsprechende Flag in der Eigenschaft Ack_Required des Notification-Class-Objektes, auf das die Eigenschaft Notification_Class dieses Objektes eindeutig verweist, nicht gesetzt ist (d. h. es wird keine Bestätigung erwartet)

In Objekt Multi-State-Value:

Diese Eigenschaft des Typs BACnetEventTransitionBits übergibt drei Flags, die den Erhalt der Bestätigung für TO-OFFNORMAL-, TO-FAULT- und TO-NORMAL-Ereignisse anzeigen. Die Flags werden nach Auftreten des entsprechenden Ereignisses gelöscht und bei folgenden Bedingungen gesetzt:

- bei Erhalt der entsprechenden Bestätigung
- bei Auftreten des Ereignisses, falls das entsprechende Flag in der Eigenschaft Event_Enable nicht gesetzt ist (d. h. für diesen Zustand werden keine Ereignisbenachrichtigungen erstellt und folglich wird keine Bestätigung erwartet)
- bei Auftreten des Ereignisses, falls das entsprechende Flag in der Eigenschaft Event_Enable gesetzt ist und das entsprechende Flag in der Eigenschaft Ack_Required des Notification-Class-Objektes, auf das die Eigenschaft Notification_Class dieses Objektes eindeutig verweist, nicht gesetzt ist (d.h. es wird keine Bestätigung erwartet)

Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt.

In Objekt Pulse-Converter:

Diese Eigenschaft des Typs BACnetEventTransitionBits übergibt drei Flags, die den Erhalt der Bestätigung für TO-OFFNORMAL-, TO-FAULT- und TO-NORMAL-Ereignisse anzeigen. Im Kontext von Pulse-Converter-Objekten werden Wechsel in die Event_States High_Limit und Low_Limit als OFFNORMAL-Ereignisse angesehen. Die Flags werden nach Auftreten des entsprechenden Ereignisses gelöscht und bei folgenden Bedingungen gesetzt:

- bei Erhalt der entsprechenden Bestätigung
- bei Auftreten des Ereignisses, falls das entsprechende Flag in der Eigenschaft Event_Enable nicht gesetzt ist (d. h. für diesen Zustand werden keine Ereignisbenachrichtigungen erstellt und folglich wird keine Bestätigung erwartet)
- bei Auftreten des Ereignisses, falls das entsprechende Flag in der Eigenschaft Event_Enable gesetzt ist und das entsprechende Flag in der Eigenschaft Ack_Required des Notification-Class-Objektes, auf das die Eigenschaft Notification_Class dieses Objektes eindeutig verweist, nicht gesetzt ist (d. h. es wird keine Bestätigung erwartet)

Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt.

In Objekt Trend-Log:

Diese Eigenschaft des Typs BACnetEventTransitionBits übergibt drei Flags, die den Erhalt der Bestätigung für TO-OFFNORMAL-, TO-FAULT- und TO-NORMAL-Ereignisse anzeigen. Die Flags werden nach Auftreten des entsprechenden Ereignisses gelöscht und bei folgenden Bedingungen gesetzt:

- bei Erhalt der entsprechenden Bestätigung
- bei Auftreten des Ereignisses, falls das entsprechende Flag in der Eigenschaft Event_Enable nicht gesetzt ist (d. h. für diesen Zustand werden keine Ereignisbenachrichtigungen erstellt und folglich wird keine Bestätigung erwartet)
- bei Auftreten des Ereignisses, falls das entsprechende Flag in der Eigenschaft Event_Enable gesetzt ist und das entsprechende Flag in der Eigenschaft Ack_Required des Notification-Class-Objektes, auf das die Eigenschaft Notification_Class dieses Objektes eindeutig verweist, nicht gesetzt ist (d. h. es wird keine Bestätigung erwartet)

Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt.

7.4 Action

Datentyp siehe Objekt¹

Datentyp IEC siehe Objekt²

In Objekt Command (BACnetARRAY [N] of BACnetActionList¹, ARRAY [1..3] of BACnetActionList²)

Beschreibung In Objekt Command:

Diese Eigenschaft des Typs „BACnetARRAY of BACnetActionList“ gibt ein Feld mit Aktionslisten an. Die Aktionslisten werden mit dem Wert der Eigenschaft Present_Value indiziert. Eine Liste kann auch leer sein, in diesem Fall wird keine Aktion ausgeführt. Es werden lediglich In_Process zurück auf „FALSE“ und All_Writes_Successful auf „TRUE“ gesetzt. Ist die Liste nicht leer, schreibt das Command-Objekt für jede Aktion in der Liste einen bestimmten Wert in eine bestimmte Eigenschaft eines Objektes in einem BACnet-Gerät, basierend auf den Spezifikationen des jeweiligen BACnetActionCommand. Jeder Schreibvorgang erfolgt in der Reihenfolge der Elemente in der Liste BACnetActionCommand, die bei Verwendung des Dienstes ReadProperty ausgelesen werden. Der Wert „Null“ ist ein Sonderfall, bei dem keine Aktion erfolgt und das Verhalten einer leeren Liste entspricht. Die Anzahl der definierten Aktionslisten kann durch Auslesen der Eigenschaft Property mit einem Feldindex von Null ermittelt werden.

Ändert sich die Größe dieses Feldes, wird die Größe des Feldes Action_Text, falls vorhanden, auf dieselbe Größe geändert. Jeder BACnetActionCommand gibt einen einzelnen Wert an, der in eine einzelne Eigenschaft eines einzelnen Objektes geschrieben wird. BACnetActionCommands bestehen aus neun Teilen: optionaler „BACnet Device Identifier“, „Object Identifier“, „Property Identifier“, bedingter „Property Array Index“, zu schreibender Wert, bedingter Priorität, optionaler Verzögerung nach dem Schreiben, Flag für vorzeitigen Abbruch und Flag für die Anzeige eines erfolgreichen Schreibvorgangs.

Die Komponenten und ihre Datentypen sind nachfolgend dargestellt:

Tabelle 69: Komponenten und Datentypen der BACnetActionCommand

Komponente	Datentyp
Device_Identifier	BACnetObjectIdentifier (optional)
Object_Identifier	BACnetObjectIdentifier
Property_Identifier	BACnetPropertyIdentifier
Property_Array_Index	Unsigned (bedingt)
Property_Value	beliebig
Priority	Unsigned (1..16) (bedingt)
Post_Delay	Unsigned (optional)
Quit_On_Failure	BOOLEAN
Write_Successful	BOOLEAN

Ist kein Device_Identifier vorhanden, wird der Schreibvorgang mit Objekten in dem Gerät ausgeführt, welches das Command-Objekt enthält. Ein Gerät, welches das Command-Objekt unterstützt, muss keine Schreibvorgänge außerhalb des Gerätes unterstützen. Verweist der Property_Identifier auf eine Feldeigenschaft, muss der Property_Array_Index vorhanden sein, um den Index des Feldes der zu schreibenden Eigenschaft anzugeben. Kann die zu schreibende Eigenschaft Befehle empfangen, wird ein Prioritätswert übergeben, andernfalls ist dieser nicht erforderlich. Ist das Flag Quit_On_Failure gleich „TRUE“, beendet das Gerät die Ausführung der Aktionsliste vorzeitig, falls der Schreibvorgang aus einem beliebigen Grund fehlschlägt. Anderfalls wird der Schreibvorgang bei einem Fehler mit dem nächsten Element der Aktionsliste fortgesetzt. In jedem Fall ist All_Writes_Successful bei Ausführung der Liste gleich „FALSE“. Nach jedem Schreibvorgang, erfolgreich oder nicht, zeigt Post_Delay, falls vorhanden, die Verzögerung bis zur Ausführung des nächsten Schreibvorganges oder bis zum Abschluss aller Schreibvorgänge und Setzen von In_Process auf „FALSE“ in Sekunden an.

Schlägt der Schreibvorgang aus irgendeinem Grund fehl, wird Write_Successful auf „FALSE“ gesetzt. Ist das Flag Quit_On_Failure gleich „TRUE“, beendet das Gerät die Ausführung der Aktionsliste vorzeitig, falls der Schreibvorgang fehlschlägt. In diesem Fall wird das Flag Write_Successful weiterer Einträge derselben Liste auf „FALSE“ gesetzt. Ist der Schreibvorgang erfolgreich, wird das Write_Successful Flag auf „TRUE“ gesetzt.

7.5 Action_Text

Datentyp	BACnetARRAY[N] of CharacterString
Datentyp IEC	ARRAY [1..3] of STRING
In Objekt	Command
Beschreibung	Diese Eigenschaft des Typs „BACnetARRAY of CharacterString“ wird für eine Textbeschreibung jedes möglichen Wertes der Eigenschaft Present_Value verwendet. Der Inhalt dieser Zeichenkette unterliegt keinen Einschränkungen. Bei Änderung der Größe dieses Feldes wird das Feld Action auf dieselbe Größe geändert.

7.6 Active_Text

Datentyp	CharacterString
Datentyp IEC	STRING
In Objekt	Binary-Value
Beschreibung	Diese Eigenschaft des Typs CharacterString beschreibt die gewünschten Auswirkungen des ACTIVE-Zustandes des Binary-Value. Der Inhalt dieser Zeichenkette ist lokal unterschiedlich, soll jedoch eine für den Anwender lesbare Beschreibung des ACTIVE-Zustandes enthalten. Wenn eine der optionalen Eigenschaften Active_Text oder Inactive_Text vorhanden ist, müssen beide verwendet werden.

7.7 Adjust_Value

Datentyp REAL

Datentyp IEC REAL

In Objekt Pulse-Converter

Beschreibung Durch Schreiben dieser Eigenschaft des Typs REAL wird die Eigenschaft Present_Value (und folglich die Eigenschaft Count) um den in Adjust_Value angegebenen Wert geändert. Ist diese Eigenschaft schreibbar, werden die nachfolgenden Aktionen bei Schreiben der Eigenschaft gemeinsam durchgeführt:

- Der in Adjust_Value geschriebene Wert wird in der Eigenschaft Adjust_Value gespeichert.
- Der Wert von Count wird in die Eigenschaft Count_Before_Change kopiert.
- Der Wert von Count wird um den durch eine Integer-Teilung ($\text{Adjust\_Value} / \text{Scale\_Factor}$) berechneten Wert verringert, der Rest wird verworfen.
- Die aktuelle Zeit und das Datum werden in der Eigenschaft Count_Change_Time gespeichert.

Schreiben in diese Eigenschaft führt zu einer Änderung des Wertes von Present_Value. Die Berechnung des neuen Wertes als Teil der Atomic-Aktionen oder durch Auslesen von Present_Value ist lokal unterschiedlich. Wurde Adjust_Value nicht geschrieben, hat die Eigenschaft den Wert „Null“.

7.8 Alarm_Value

Datentyp BACnetBinaryPV

Datentyp IEC BACnetBinaryPV

In Objekt Binary-Value

Beschreibung Diese Eigenschaft des Typs BACnetBinaryPV gibt den für Present_Value erforderlichen Wert vor Generierung eines OFFNORMAL-Ereignisses an. Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt.

7.9 Alarm_Values

Datentyp siehe Objekt¹

Datentyp IEC siehe Objekt²

In Objekt Life-Safety-Point/Life-Safety-Zone (List of BACnetLifeSafetyState1, ARRAY [0..3] of BACnetLifeSafetyState²)
Multi-State-Value (List of Unsigned1, ARRAY [1..3] of WORD²)

Beschreibung **In Objekt Life-Safety-Point:**
Diese optionale Eigenschaft des Typs „List of BACnetLifeSafetyState“ gibt Zustände an, die der Present_Value annehmen muss, bevor ein TO-OFFNORMAL-Ereignis generiert und der OFFNORMAL-Zustand aufgerufen wird. Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt.

Bedingungen für die Generierung eines TO-OFFNORMAL-Ereignisses:

- das Flag TO-OFFNORMAL muss in der Eigenschaft Event_Enable aktiviert sein und
- der Present_Value muss einem der Werte in der Liste Alarm_Values entsprechen und
- der Present_Value muss den in der Liste Alarm_Values angegebenen Wert für den in der Eigenschaft Time_Delay angegebenen Zeitraum behalten.

Neue Ereignisse werden bei einer Änderung von Mode generiert, falls das Flag TO-OFFNORMAL in der Eigenschaft Event_Enable gesetzt ist.

Die Bedingungen für die Generierung eines TO-NORMAL-Ereignisses sind in Kapitel „Life_Safety_Alarm_Values“ definiert.

In Objekt Multistate Input:

Diese Eigenschaft des Typs „List of Unsigned“ gibt Zustände an, denen der Present_Value entsprechen muss, bevor ein TO-OFFNORMAL-Ereignis generiert wird. Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt. Bedingungen für die Generierung eines TO-OFFNORMAL-Ereignisses:

- der Present_Value muss mindestens einem der Werte in der Liste Alarm_Values entsprechen und
- der Present_Value muss für den in der Eigenschaft Time_Delay festgelegten Mindestzeitraum denselben Wert beibehalten und
- das Flag TO-OFFNORMAL muss in der Eigenschaft Event_Enable aktiviert sein.

Bedingungen für die Generierung eines TO-NORMAL-Ereignisses:

Ist der Wert einmal gleich, darf der Present_Value keinen Zustand dieser Eigenschaft und keinen Zustand der Eigenschaft Fault_Values annehmen, bevor aufgrund folgender Bedingungen ein TO-NORMAL-Ereignis generiert wird:

- der Present_Value darf für den in der Eigenschaft Time_Delay festgelegten Mindestzeitraum keinem in der Liste Alarm_Values enthaltenen Wert entsprechen und
- der Present_Value darf keinem in der Eigenschaft Fault_Values enthaltenen Wert entsprechen und
- das Flag TO-NORMAL muss in der Eigenschaft Event_Enable aktiviert sein.

In Objekt Multi-State-Value:

Diese Eigenschaft des Typs „List of Unsigned“ gibt Zustände an, denen der Present_Value entsprechen muss, bevor ein TO-OFFNORMAL-Ereignis generiert wird. Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt. Bedingungen für die Generierung eines TO-OFFNORMAL-Ereignisses:

- der Present_Value muss mindestens einem Wert in der Liste Alarm_Values entsprechen und
- der Present_Value muss für den Mindestzeitraum, der in der Eigenschaft Time_Delay festgelegt ist, denselben Wert beibehalten und
- das Flag TO-OFFNORMAL muss in der Eigenschaft Event_Enable aktiviert sein.

Bedingungen für die Generierung eines TO-NORMAL-Ereignisses:

Ist der Wert einmal gleich, darf der Present_Value keinen Zustand dieser Eigenschaft und keinen Zustand der Eigenschaft Fault_Values annehmen, bevor aufgrund folgender Bedingungen ein TO-NORMAL-Ereignis generiert wird:

- der Present_Value darf für den in der Eigenschaft Time_Delay festgelegten Mindestzeitraum keinem Wert in der Liste Alarm_Values Wert entsprechen und
- der Present_Value darf keinem Wert in der Eigenschaft Fault_Values entsprechen und
- das Flag TO-NORMAL muss in der Eigenschaft Event_Enable aktiviert sein.

7.10 All_Writes_Successful

Datentyp BOOLEAN

Datentyp IEC BOOL

In Objekt Command

Beschreibung Diese Eigenschaft des Typs BOOLEAN gibt Erfolg oder Fehlschlagen der beim Schreiben in die Eigenschaft Present_Value ausgelösten Aktionen an. Gleichzeitig wird In_Process auf „TRUE“ und All_Writes_Successful auf „FALSE“ gesetzt. Waren nach Ausführen der gesamten Liste alle Schreibvorgänge erfolgreich, wird All_Writes_Successful auf „TRUE“ und gleichzeitig In_Process auf „FALSE“ gesetzt. Folglich stellt All_Writes_Successful keine gültige Angabe der aktuellen oder der letzten Aktion dar, solange In_Process gleich „TRUE“ ist.

7.11 Archive

Datentyp BOOLEAN

Datentyp IEC BOOL

In Objekt File

Beschreibung Diese Eigenschaft des Typs BOOLEAN gibt an, ob das File-Objekt für die Verlaufs- oder Datensicherung gespeichert wurde. Die Eigenschaft darf nur logisch „TRUE“ sein, wenn die Dateidaten seit der letzten Archivierung durch einen internen Prozess oder über Dienste für den Dateizugriff nicht verändert wurden.

7.12 Attempted_Samples

Datentyp Unsigned

Datentyp IEC WORD

In Objekt Averaging

Beschreibung Diese Eigenschaft des Typs Unsigned gibt die Anzahl der Samples an, die für das aktuelle Fenster verwendet werden sollten. In diese Eigenschaft darf nur der Wert „Null“ geschrieben werden. Falls Attempted_Samples geringer ist als Window_Samples, dann ist eine Zeitspanne geringer als Window_Interval seit Neustart des Gerätes oder seit Schreiben von Attempted_Samples, Object_Property_Reference, Window_Samples oder Window_Interval mit BACnet-Diensten vergangen. Die Anzahl der verpassten Samples des aktuellen Fensters kann durch Abzug von Valid_Samples von Attempted_Samples berechnet werden. Nach einem Neustart des Gerätes oder nach dem Schreiben von Attempted_Samples, Object_Property_Reference, Window_Samples oder Window_Interval mit BACnet-Diensten hat Attempted_Samples den Wert „Null“, bis ein Sampling erfolgt.

7.13 Average_Value

Datentyp REAL

Datentyp IEC REAL

In Objekt Averaging

Beschreibung Diese Eigenschaft des Typs REAL gibt den Durchschnittswert des Speicherfensters für die aktuellsten Window_Samples oder die aktuelle Sample-Anzahl (Valid_Samples) an, falls weniger Samples als Window_Samples vorhanden sind. Der Durchschnitt wird durch Teilen der arithmetischen Summe aller erfolgreichen Samples im Speicher durch die Sample-Anzahl berechnet. Nach einem Neustart des Gerätes oder nach dem Schreiben von Attempted_Samples, Object_Property_Reference, Window_Samples oder Window_Interval mit BACnet-Diensten hat Average_Value den Wert NaN, bis ein Sampling erfolgt.

7.14 Buffer_Size

Datentyp Unsigned32

Datentyp IEC DWORD

In Objekt Trend-Log

Beschreibung Diese Eigenschaft des Typs Unsigned32 gibt die maximale Anzahl von Einträgen an, die der Speicher enthalten kann. Falls schreibbar ist das Schreiben unter Umständen nicht möglich, wenn Log_Enable gleich „TRUE“ ist. Die Handhabung bestehender Einträge beim Schreiben der Buffer_Size ist lokal unterschiedlich.

7.15 Change_Of_State_Count

Datentyp Unsigned32

Datentyp IEC DWORD

In Objekt Binary-Value

Beschreibung Diese Eigenschaft des Typs Unsigned32 gibt an, wie oft sich der Zustand von Binary-Value seit dem letzten Zurücksetzen dieser Eigenschaft geändert hat. Die Eigenschaft Change_Of_State_Count hat einen Wert zwischen „0“ und 65535 oder höher. Falls eine der optionalen Eigenschaften Change_Of_State_Time, Change_Of_State_Count oder Time_Of_State_Count_Reset verwendet wird, müssen alle diese Eigenschaften vorhanden sein.

7.16 Change_Of_State_Time

Datentyp BACnetDateTime

Datentyp IEC BACnetDateTime

In Objekt Binary-Value

Beschreibung Diese Eigenschaft des Typs BACnetDateTime gibt Zeit und Datum der letzten Zustandsänderung an. Als Zustandsänderung (Change of State) gilt jedes Ereignis, das den logischen Zustand von Binary-Value verändert. Falls eine der optionalen Eigenschaften Change_Of_State_Time, Change_Of_State_Count oder Time_Of_State_Count_Reset vorhanden ist, müssen alle diese Eigenschaften vorhanden sein.

7.17 Client_COV_Increment

Datentyp BACnetClientCOV

Datentyp IEC BACnetClientCOV

In Objekt Trend-Log

Beschreibung Falls mittels Trend-Log COV-Daten erfasst werden, gibt diese Eigenschaft des Typs BACnetClientCOV die für die Bestimmung einer Werteänderung erforderlichen Schwellwerte an. Unterstützen die referenzierte Eigenschaft und das Objekt COV-Reporting gemäß Kapitel „Melden bei Wertänderungen (Change Of Value / State Reporting)“, kann diese Eigenschaft den Wert „Null“ haben. In diesem Fall wird eine Werteänderung anhand der Kriterien in dem Kapitel bestimmt.

7.18 Count

Datentyp Unsigned

Datentyp IEC WORD

In Objekt Pulse-Converter

Beschreibung Diese nur lesbare Eigenschaft des Typs Unsigned gibt die Anzahl der Impulse des physikalischen Einganges oder der Eigenschaft an, auf die Input_Reference verweist. Ist die Eigenschaft, auf die Input_Reference verweist, vorhanden, hat den Datentyp Unsigned oder Integer und wird als Eingangsquelle für dieses Objekt unterstützt, wird der Wert der Eigenschaft Count von der referenzierten Eigenschaft abgeleitet. Eine Veränderung von „1“ in der referenzierten Eigenschaft wird durch eine Veränderung des Count um „1“ angezeigt. Das Verfahren hierfür ist lokal unterschiedlich. Da der Wert der Eigenschaft Count des Pulse-Converter-Objektes durch Schreiben in die Eigenschaft Adjust_Value geändert werden kann, ist es möglich, dass sich der Wert der Eigenschaft Count vom Wert der referenzierten Eigenschaft unterscheidet.

7.19 Count_Before_Change

Datentyp Unsigned

Datentyp IEC WORD

In Objekt Pulse-Converter

Beschreibung Diese Eigenschaft des Typs Unsigned gibt den Wert der Eigenschaft Count vor dem letzten Schreiben in die Eigenschaft Adjust_Value an. Ist noch kein Schreibvorgang erfolgt, hat diese Eigenschaft den Wert „Null“.

7.20 Count_Change_Time

Datentyp BACnetDateTime

Datentyp IEC BACnetDateTime

In Objekt Pulse-Converter

Beschreibung Diese nur lesbare Eigenschaft des Typs BACnetDateTime gibt Zeit und Datum des letzten Schreibvorgangs in die Eigenschaft Adjust_Value an. Ist noch kein Schreibvorgang erfolgt, hat diese Eigenschaft Platzhalter für alle Datums- und Zeitfelder.

7.21 COV_Increment

Datentyp REAL

Datentyp IEC REAL

In Objekt Analog-Value, Pulse-Converter

Beschreibung Diese Eigenschaft des Typs REAL gibt die erforderliche Mindestveränderung des Present_Value an, bei der eine COVNotification an hierfür konfigurierte COV-Clients gesendet wird. Diese Eigenschaft ist erforderlich, falls das Objekt COV-Reporting unterstützt.

7.22 COV_Period

Datentyp Unsigned

Datentyp IEC WORD

In Objekt Pulse-Converter

Beschreibung Die Eigenschaft COV_Period des Typs Unsigned gibt die Zeit in Sekunden zwischen regelmäßigen COV-Benachrichtigungen für dieses Objekt an. Diese Eigenschaft ist erforderlich, falls das Objekt COV-Reporting unterstützt.

7.23 COV_Resubscription_Interval

Datentyp Unsigned

Datentyp IEC WORD

In Objekt Trend-Log

Beschreibung Wenn Trend-Log-Daten eines entfernten Gerätes per COV-Abonnement erfasst werden, gibt diese Eigenschaft des Typs Unsigned die Zeit in Sekunden zwischen COV-Neu-Abonnements an, falls bereits ein COV-Abonnement vorliegt. SubscribeCOV-Anfragen geben das doppelte dieser Zeit für ein Abonnement an und ob eine Bestätigung der Benachrichtigung erforderlich ist oder nicht. Liegen COV-Abonnements vor, wird die erste COV-Benachrichtigung ausgegeben, wenn das Trend-Log-Objekt mit der Aufzeichnung beginnt oder wenn Log_Enable auf „TRUE“ gesetzt wird. Falls vorhanden, darf der Wert dieser Eigenschaft nicht Null sein. Ist diese Eigenschaft nicht vorhanden, wird kein COV-Abonnement versucht.

7.24 Date_List

Datentyp	List of BACnetCalendarEntry
Datentyp IEC	ARRAY [0...3] of BACnetCalendarEntry;
In Objekt	Calendar
Beschreibung	Diese Eigenschaft ist eine „List of BACnetCalendarEntry“. Jeder Eintrag ist entweder ein bestimmtes Datum (Date), ein Datumsbereich (BACnetDateRange) oder eine Monats-/Wochen-/Tagesangabe (BACnetWeekNDay). Stimmt das aktuelle Datum mit den Kriterien des Kalendereintrages überein, ist der Wert des Calendar-Objektes gleich „TRUE“. Einzelne Felder der verschiedenen Konstrukte können auch leer bleiben. In diesem Fall dient das Feld als Platzhalter für die Bestimmung aktueller Datumsergebnisse einer Übereinstimmung. Wenn beispielsweise bei einem Datumsbereich das Startdatum (startDate) leer ist, bedeutet dies „beliebiges Datum bis einschließlich endDate“. Ist kein Enddatum (endDate) angegeben bedeutet dies „ein beliebiges Datum ab startDate“. Ist der Kalendereintrag BACnetWeekNDay ein bestimmter Wochentag ohne Monats-/Wochenangabe, bedeutet dies, dass das Kalenderobjekt an diesem Wochentag das ganze Jahr über „TRUE“ ist. Wenn ein BACnet-Gerät das Schreiben in die Eigenschaft Date_List zulässt, sind alle Auswahlmöglichkeiten in BACnetCalendarEntry zulässig.

7.25 Deadband

Datentyp REAL

Datentyp IEC REAL

In Objekt Analog-Value, Pulse-Converter

Beschreibung Diese Eigenschaft des Typs REAL gibt den Bereich zwischen den Eigenschaften High_Limit und Low_Limit an, in dem der Wert Present_Value für ein TO-NORMAL-Ereignis verbleiben muss, das unter folgenden Bedingungen generiert wird:

- der Present_Value fällt unter das High_Limit minus Deadband und Present_Value übersteigt das Low_Limit plus Deadband und
- der Present_Value verbleibt für mindestens den in der Eigenschaft Time_Delay angegebenen Zeitraum in diesem Bereich und
- das Flag HighLimitEnable oder LowLimitEnable muss in der Eigenschaft Limit_Enable aktiviert sein und
- das Flag TO-NORMAL muss in der Eigenschaft Event_Enable aktiviert sein.

Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt.

7.26 Description

Datentyp CharacterString

Datentyp IEC STRING

In Objekt Accumulator, Analog-Value, Averaging, Binary-Value, Calendar, Command, File, Group, Life-Safety-Point, Life-Safety-Zone, Multi-State-Value, Notification-Class, Pulse-Converter, Schedule, Trend-Log

Beschreibung Diese Eigenschaft des Typs CharacterString ist eine Zeichenkette druckbarer Zeichen ohne Inhaltsbeschränkungen.

7.27 Device_Type

Datentyp `CharacterString`

Datentyp IEC `STRING`

In Objekt Accumulator, Life-Safety-Point, Life-Safety-Zone

Beschreibung **In Objekt Accumulator:**
Diese Eigenschaft des Typs `CharacterString` ist eine Textbeschreibung des vom Accumulator-Objekt dargestellten physikalischen Gerätes. Die Eigenschaft wird in der Regel verwendet, um den von Accumulator dargestellten Sensor zu beschreiben.

In Objekt Life-Safety-Point:
Diese optionale Eigenschaft des Typs `CharacterString` ist eine Textbeschreibung des vom Life-Safety-Point-Objekt dargestellten physikalischen Gerätes.

In Objekt Life-Safety-Zone:
Diese optionale Eigenschaft des Typs `CharacterString` ist eine Textbeschreibung des vom Life-Safety-Zone-Objekt dargestellten physikalischen Bereichs. Die Eigenschaft wird in der Regel für die Beschreibung des Standortes der Life-Safety-Point-Objekte beschrieben, die `Zone_Member` des Life-Safety-Zone-Objektes sind.

7.28 Direct_Reading

Datentyp REAL

Datentyp IEC REAL

In Objekt Life-Safety-Point

Beschreibung Diese optionale Eigenschaft des Typs REAL gibt einen Analogwert an, der den Messwert oder den berechneten Wert eines initiierenden Gerätes darstellt. Die Methode für die Verwendung dieses Werts zur Bestimmung eines logischen Zustandes des Gerätes ist lokal unterschiedlich. Ist diese Eigenschaft vorhanden, muss auch die Eigenschaft Units verwendet werden.

7.29 Effective_Period

Datentyp BACnetDateRange

Datentyp IEC BACnetDateRange

In Objekt Schedule

Beschreibung Diese Eigenschaft gibt den Datumsbereich an, in dem das Schedule-Objekt aktiv ist. Saisonale Zeitplanung ist unter Umständen durch die Definition mehrerer Schedule-Objekte für die Steuerung derselben Eigenschaftsreferenzen möglich, deren Effective_Periods sich nicht überschneiden. Nach Eingabe des Gültigkeitszeitraums berechnet das Objekt seinen Present_Value und übergibt diesen Wert an alle Mitglieder der Eigenschaft List_Of_Object_Property_References. Bei einem Fehler bei der Übergabe an ein Listenmitglied fährt das Schedule-Objekt mit der Übergabe an die verbleibenden Mitglieder fort.

7.30 Elapsed_Active_Time

Datentyp Unsigned32

Datentyp IEC DWORD

In Objekt Binary-Value

Beschreibung Diese Eigenschaft des Typs Unsigned32 gibt die Gesamtzeit in Sekunden an, welche die Eigenschaft Present_Value seit dem letzten Zurücksetzen der Eigenschaft Elapsed_Active_Time den Wert ACTIVE hatte. Wenn eine der optionalen Eigenschaften Elapsed_Active_Time oder Time_Of_Active_Time_Reset vorhanden ist, müssen beide verwendet werden.

7.31 Event_Enable

Datentyp BACnetEventTransitionBits

Datentyp IEC BACnetEventTransitionBits

In Objekt Accumulator, Analog-Value, Binary-Value, Life-Safety-Point, Life-Safety-Zone, Multi-State-Value, Pulse-Converter, Trend-Log

Beschreibung **In Objekt Accumulator:**
Diese Eigenschaft des Typs BACnetEventTransitionBits übergibt drei Flags, welche die Meldung von TO-OFFNORMAL-, TO-FAULT- und TO-NORMAL-Ereignissen aktivieren bzw. deaktivieren. Im Kontext von Accumulator-Objekten werden Wechsel in die Event_States High_Limit und Low_Limit als OFFNORMAL-Ereignisse angesehen. Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt.

In Objekt Analog-Value:
Diese Eigenschaft des Typs BACnetEventTransitionBits übergibt drei Flags, welche die Meldung von TO-OFFNORMAL-, TO-FAULT- und TO-NORMAL-Ereignissen aktivieren bzw. deaktivieren. Im Kontext von Analog-Value-Objekten werden Wechsel in die Event_States High_Limit und Low_Limit als OFFNORMAL-Ereignisse angesehen. Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt.

In Objekt Binary-Value:

Diese Eigenschaft des Typs BACnetEventTransitionBits übergibt drei Flags, welche die Meldung von TO-OFFNORMAL-, TO-FAULT- und TO-NORMAL-Ereignissen aktivieren bzw. deaktivieren. Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt.

In Objekt File:

Diese optionale Eigenschaft des Typs BACnetEventTransitionBits übergibt drei Flags, welche die Meldung von TO-OFFNORMAL-, TO-FAULT- und TO-NORMAL-Ereignissen aktivieren bzw. deaktivieren, die auf Änderungen des Present_Value und/oder Mode basieren. Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt.

In Objekt Life-Safety-Zone:

Diese optionale Eigenschaft des Typs BACnetEventTransitionBits übergibt drei Flags, welche die Meldung von TO-OFFNORMAL-, TO-FAULT- und TO-NORMAL-Ereignissen aktivieren bzw. deaktivieren, die auf Änderungen des Present_Value und/oder des Mode basieren.

In Objekt Multi-State-Value:

Diese Eigenschaft des Typs BACnetEventTransitionBits übergibt drei Flags, welche die Meldung von TO-OFFNORMAL-, TO-FAULT- und TO-NORMAL-Ereignissen aktivieren bzw. deaktivieren. Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt.

In Objekt Pulse-Converter:

Diese Eigenschaft des Typs BACnetEventTransitionBits übergibt drei Flags, welche die Meldung von TO-OFFNORMAL-, TO-FAULT- und TO-NORMAL-Ereignissen aktivieren bzw. deaktivieren. Im Kontext von Pulse-Converter-Objekten werden Wechsel in die Event_States High_Limit und Low_Limit als OFFNORMAL-Ereignisse angesehen. Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt.

In Objekt Trend-Log:

Diese Eigenschaft des Typs BACnetEventTransitionBits übergibt drei Flags, welche die Meldung von TO-OFFNORMAL-, TO-FAULT- und TO-NORMAL-Ereignissen aktivieren bzw. deaktivieren. Im Kontext von Trend-Log-Objekten gilt es als TO-NORMAL-Ereignis, wenn die Eigenschaft Records_Since_Notification gleich oder größer dem Wert der Eigenschaft Notification_Threshold ist. Das Fehlschlagen eines COV-Abonnementversuchs gilt als TO-FAULT-Ereignis. Das TO-NORMAL-Ereignis muss aktiviert sein, falls objektinternes Melden verwendet werden soll. Dies ist die Standardeinstellung. Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt.

7.32 Event_State

Datentyp BACnetEventState

Datentyp IEC BACnetEventState

In Objekt Accumulator, Analog-Value, Binary-Value, Life-Safety-Point, Life-Safety-Zone, Multi-State-Value, Pulse-Converter, Trend-Log

Beschreibung **Allgemein:**
Die Eigenschaft Event_State des Typs BACnetEventState bietet eine Möglichkeit zur Bestimmung, ob dem Objekt ein aktiver Ereigniszustand zugewiesen ist. Unterstützt das Objekt objektinternes Melden, muss die Eigenschaft Event_State den Ereigniszustand des Objektes angeben.

In Objekt Accumulator, Analog-Value und Binary-Value:

Unterstützt das Objekt objektinternes Melden nicht (und bei einem Objekt ist die Eigenschaft Reliability nicht vorhanden), ist der Wert dieser Eigenschaft NORMAL. Ist die Eigenschaft Reliability vorhanden und hat nicht den Wert NO_FAULT_DETECTED, ist der Wert der Eigenschaft Event_State FAULT. Eine Änderung der Eigenschaft Event_State auf den Wert FAULT wird als Fault-Ereignis angesehen.

In Objekt Multi-State-Value:

Wenn das Objekt objektinternes Melden nicht unterstützt:

- ist die Eigenschaft Reliability nicht vorhanden, ist der Wert von Event_State NORMAL oder
- ist die Eigenschaft Reliability vorhanden und Reliability ist NO_FAULT_DETECTED, ist der Event_State NORMAL oder
- ist die Eigenschaft Reliability vorhanden und Reliability ist nicht NO_FAULT_DETECTED, ist der Event_State FAULT.

In Objekt Life-Safety-Point:

Unterstützt das Objekt kein objektinternes Melden, ist der Wert der Eigenschaft NORMAL.

In Objekt Pulse-Converter:

Unterstützt das Objekt kein objektinternes Melden und die Eigenschaft Reliability ist nicht vorhanden, ist der Wert dieser Eigenschaft NORMAL. Ist die Eigenschaft Reliability vorhanden und hat nicht den Wert NO_FAULT_DETECTED, ist der Wert der Eigenschaft Event_State FAULT. Eine Änderung der Eigenschaft Event_State auf den Wert FAULT wird als Fault-Ereignis angesehen.

In Objekt Trend-Log:

Unterstützt das Objekt kein objektinternes Melden, ist der Wert der Eigenschaft NORMAL. Gültige Zustände sind NORMAL und FAULT.

7.33 Event_Time_Stamps

Datentyp	BACnetARRAY [3] of BACnetTimeStamp
Datentyp IEC	ARRAY [1...3] of BACnetTimeStamp
In Objekt	Accumulator, Analog-Value, Binary-Value, Life-Safety-Point, Life-Safety-Zone, Multi-State-Value, Pulse-Converter, Trend-Log
Beschreibung	Diese optionale Eigenschaft des Typs „BACnetARRAY [3] of BACnetTimeStamp“ gibt die Zeiten der letzten Ereignisbenachrichtigungen für TO-OFFNORMAL-, TO-FAULT- und TO-NORMAL-Ereignisse an. Zeitstempel des Typs Time oder Date müssen „FF“ in jedem Oktett enthalten, Zeitstempel mit Sequenznummer müssen den Wert „0“ aufweisen, wenn seit Erstellung des Objektes keine Ereignisbenachrichtigung dieses Typs generiert wurde. Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt.

7.34 Exception_Schedule

Datentyp	BACnetARRAY [N] of BACnetSpecialEvent
Datentyp IEC	ARRAY [1...7] of BACnetSpecialEvent
In Objekt	Schedule
Beschreibung	Diese Eigenschaft ist ein „BACnetARRAY of BACnetSpecialEvents“. Jedes BACnetSpecialEvent beschreibt eine Abfolge von Schedule-Aktionen, die an einem oder mehreren bestimmten Tagen Vorrang vor dem normalen Verhalten für diesen Tag haben.

BACnetSpecialEvent ::= (Period, List of BACnetTimeValue, EventPriority)

Period ::= Auswahl zwischen {BACnetCalendarEntry | CalendarReference}

EventPriority ::= Unsigned (1...16)

Period kann ein BACnetCalendarEntry sein oder auf ein Kalenderobjekt verweisen. Ein BACnetCalendarEntry wird verwendet, wenn Exception_Schedule nur für dieses Schedule-Objekt gilt, ein Kalender kann beispielsweise verwendet werden, wenn mehrere Schedule-Objekte auf Feiertage verweisen. Jeder BACnetCalendarEntry ist entweder ein bestimmtes Datum (Date), ein Datumsbereich (BACnetDateRange) oder eine Monats-/Wochen-/Tagesangabe (BACnetWeekNDay). Stimmt das aktuelle Datum mit den Kriterien des Kalendereintrages überein, wird die Eigenschaft Exception_Schedule aktiviert und die „List of BACnetTimeValues“ steht zur Verfügung.

Einzelne Felder der verschiedenen Konstrukte für BACnetCalendarEntry bieten unter Umständen Platzhalter zur Bestimmung, ob das aktuelle Datum im Zeitraum der Eigenschaft Exception_Schedule liegt. Ist bei einem Datumsbereich, z. B. startDate, ein Platzhalter, bedeutet dies „beliebiges Datum bis einschließlich endDate“. Ist endDate ein Platzhalter, bedeutet dies „ein beliebiges Datum ab startDate“.

Ist der Kalendereintrag vom Typ BACnetWeekNDay mit Platzhaltern in den Feldern Monat und Woche aber mit einem Eintrag für einen bestimmten Wochentag, bedeutet dies, dass die Eigenschaft Exception_Schedule an diesem Wochentag das ganze Jahr über gilt.

Jedes BACnetSpecialEvent enthält eine EventPriority, die das relative Verhältnis zu anderen BACnetSpecialEvents in derselben Exception_Schedule festlegt. Da sich die Zeiträume von SpecialEvents einer Exception_Schedule überschneiden können, ist ein Mechanismus zur Bestimmung der relativen Prioritäten für SpecialEvents eines Tages erforderlich. Gilt an einem Tag mehr als ein SpecialEvent, muss die relative Priorität der SpecialEvents mithilfe des Wertes EventPriority jedes Ereignisses festgelegt werden. Haben mehrere sich überschneidende SpecialEvents denselben EventPriority-Wert, hat das SpecialEvent mit der niedrigsten Indexnummer des Feldes eine höhere relative Priorität. Der höchste Wert für EventPriority ist 1, der niedrigste 16. EventPriority steht nicht im Zusammenhang mit der Eigenschaft Priority_For_Writing des Schedule-Objektes.

Falls ein BACnet-Gerät das Schreiben in die Eigenschaft Exception_Schedule unterstützt, müssen alle möglichen Varianten der BACnetSpecialEvents unterstützt werden. Steigt die Größe dieses Feldes durch Schreiben in den Feldindex Null, muss jedes neue Element des Feldes eine leere Liste für BACnetTimeValue enthalten.

Wird die Eigenschaft Exception_Schedule mit einem Schedule-Element geschrieben, das einen von dieser Instanz des Schedule-Objektes nicht unterstützten Datentyp enthält (z. B. kann die Eigenschaft List_Of_Object_Property_References nicht so konfiguriert werden, dass sie auf eine Eigenschaft eines nicht unterstützten Datentyps verweist), gibt das Gerät unter Umständen Result(-) mit der Fehlerklasse PROPERTY und dem Fehlercode DATATYPE_NOT_SUPPORTED zurück.

7.35 Fault_Values

Datentyp	siehe Objekt ¹
Datentyp IEC	siehe Objekt ²
In Objekt	Life-Safety-Point/Life-Safety-Zone (List of BACnetLifeSafetyState ¹ , ARRAY [0...3] of BACnetLifeSafetyState ²), Multi-State-Value (List of Unsigned ¹ , ARRAY [1...3] of WORD ²)
Beschreibung	In Objekt Life-Safety-Point und Life-Safety-Zone: Diese optionale Eigenschaft des Typs „List of BACnetLifeSafetyState“ gibt Zustände an, denen der Present_Value entsprechen muss, bevor ein TO-FAULT-Ereignis generiert wird. Ist Present_Value gleich einem der Zustände in der Liste Fault_Values und es wurde kein physikalischer Fehler an Eingängen erkannt, die der Present_Value darstellt, nimmt die Eigenschaft Reliability den Wert MULTI_STATE_FAULT an. Die Eigenschaft Fault_Values ist erforderlich, wenn das Objekt objektinternes Melden unterstützt. Neue Ereignisse werden bei einer Änderung von Mode generiert, falls das Flag TO-FAULT in der Eigenschaft Event_Enable gesetzt ist. In Objekt Multi-State-Value: Diese Eigenschaft des Typs „List of Unsigned“ gibt Zustände an, denen Present_Value entsprechen muss, bevor ein TO-FAULT-Ereignis generiert wird. Ist Present_Value gleich einem der Zustände in der Liste Fault_Values und es wurde kein physikalischer Fehler an Ein- oder Ausgängen erkannt, die der Present_Value darstellt, nimmt die Eigenschaft Reliability den Wert MULTI_STATE_FAULT an. Die Eigenschaft Fault_Values ist erforderlich, wenn das Objekt objektinternes Melden unterstützt.

7.36 File_Access_Method

Datentyp	BACnetFileAccessMethod
Datentyp IEC	BACnetFileAccessMethod
In Objekt	File
Beschreibung	Diese Eigenschaft des Typs BACnetFileAccessMethod gibt die Art(en) des Dateizugriffs für dieses Objekt an. Mögliche Wert für File_Access_Method sind: {RECORD_ACCESS, STREAM_ACCESS}.

7.37 File_Size

Datentyp	Unsigned
Datentyp IEC	WORD
In Objekt	File
Beschreibung	Diese Eigenschaft des Typs Unsigned gibt die Dateigröße in Oktetten an. Falls die Dateigröße durch Schreiben in die Datei mit der File_Access_Method STREAM_ACCESS verändert werden kann, muss diese Eigenschaft schreibbar sein. Das Schreiben in die Eigenschaft File_Size mit einem geringeren als dem aktuellen Wert kürzt die Datei an der angegebenen Position. Das Schreiben einer File_Size von „0“ löscht alle Daten der Datei, nicht jedoch das Dateiojekt selbst. Das Schreiben in die Eigenschaft File_Size mit einem größeren als dem aktuellen Wert erweitert die Dateigröße, der Wert der neuen Oktette der Datei ist lokal unterschiedlich. Geräte können die für das Schreiben der File_Size zulässigen Werte einschränken. Dies bedeutet insbesondere, dass Geräte das Löschen des Dateiinhaltes durch Schreiben des Wertes Null zulassen können, nicht jedoch eine beliebige Kürzung oder Erweiterung.

7.38 File_Type

Datentyp	CharacterString
Datentyp IEC	STRING
In Objekt	File
Beschreibung	Diese Eigenschaft des Typs CharacterString gibt die bestimmungsgemäße Verwendung dieser Datei an.

7.39 High_Limit

Datentyp	siehe Objekt ¹
Datentyp IEC	siehe Objekt ²
In Objekt	Accumulator (Unsigned ¹ , WORD2), Analog Value/Pulse-Converter (REAL ¹ , REAL ²)
Beschreibung	In Objekt Accumulator: Diese Eigenschaft des Typs Unsigned gibt den Grenzwert an, den Pulse_Rate vor Generierung eines Ereignisses überschreiten muss. Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt. In Objekt Analog-Value und Pulse-Converter: Diese Eigenschaft des Typs REAL gibt den Grenzwert an, den der Present_Value vor Generierung eines Ereignisses überschreiten muss. Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt.

7.40 In_Process

Datentyp BOOLEAN

Datentyp IEC BOOL

In Objekt Command

Beschreibung Diese Eigenschaft des Typs BOOLEAN wird beim Schreiben eines Wertes in die Eigenschaft Present_Value auf „TRUE“ gesetzt. Der Wert „TRUE“ gibt an, dass das Command-Objekt mit der Bearbeitung einer Aktionsliste begonnen hat. Nach Versuch der Ausführung aller Schreibvorgänge durch das Command-Objekt wird die Eigenschaft In_Process zurück auf „FALSE“ gesetzt.

7.41 Inactive_Text

Datentyp CharacterString

Datentyp IEC STRING

In Objekt Binary-Value

Beschreibung Diese Eigenschaft des Typs CharacterString beschreibt die gewünschten Auswirkungen des INACTIVE-Zustandes des Binary-Value. Der Inhalt dieser Zeichenkette ist lokal unterschiedlich, soll jedoch eine für den Anwender lesbare Beschreibung des INACTIVE-Zustandes enthalten. Wenn eine der optionalen Eigenschaften Active_Text oder Inactive_Text vorhanden ist, müssen beide verwendet werden.

7.42 Input_Reference

Datentyp	BACnetObjectPropertyReference
Datentyp IEC	BACnetObjectPropertyReference
In Objekt	Pulse-Converter
Beschreibung	Diese optionale Eigenschaft des Typs BACnetObjectPropertyReference gibt Objekt und Eigenschaft (in der Regel die Eigenschaft Present_Value eines Accumulator-Objektes) an, die den tatsächlichen physikalischen Eingang darstellen, den das Pulse-Converter-Objekt misst und angibt. Die referenzierte Eigenschaft muss den Datentyp Integer oder Unsigned haben. Ist diese Eigenschaft nicht vorhanden, stellt das Pulse-Converter-Objekt den physikalischen Eingang direkt dar.

7.43 Last_Notify_Record

Datentyp	Unsigned32
Datentyp IEC	DWORD
In Objekt	Trend-Log
Beschreibung	Diese Eigenschaft des Typs Unsigned32 gibt die dem aktuellsten Eintrag zugewiesene SequenceNumber an, dessen Ermittlung eine Benachrichtigung ausgelöst hat. Ist seit Beginn der Protokollierung dieser Eigenschaft keine Benachrichtigung erfolgt, hat diese Eigenschaft den Wert „Null“. Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt.

7.44 Life_Safety_Alarm_Values

Datentyp List of BACnetLifeSafetyState

Datentyp IEC ARRAY [0..3] of BACnetLifeSafetyState

In Objekt Life-Safety-Point, Life-Safety-Zone

Beschreibung Diese optionale Eigenschaft des Typs „List of BACnetLifeSafetyState“ gibt Zustände an, denen der Present_Value entsprechen muss, bevor ein TO-OFFNORMAL-Ereignis generiert und der Zustand LIFE_SAFETY_ALARM aufgerufen wird. Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt. Bedingungen für die Generierung eines TO-OFFNORMAL-Ereignisses:

- das Flag TO-OFFNORMAL muss in der Eigenschaft Event_Enable aktiviert sein und
- der Present_Value muss einem der in der Liste Life_Safety_Alarm_Values enthaltenen Werte entsprechen und
- der Present_Value behält für mindestens den in der Eigenschaft Time_Delay angegebenen Zeitraum einen in der Liste Life_Safety_Alarm_Values enthaltenen Wert.

Neue Ereignisse werden bei einer Änderung von Mode generiert, falls das Flag TO-OFFNORMAL in der Eigenschaft Event_Enable gesetzt ist.

Bedingungen für die Generierung eines TO-NORMAL-Ereignisses:

Ist der Wert einmal gleich, darf der Present_Value keinen nicht in der Eigenschaft Life_Safety_Alarm_Values enthaltenen, keinen nicht in der Eigenschaft Alarm_Values enthaltenen und keinen Zustand der Eigenschaft Fault_Values annehmen, bevor aufgrund folgender Bedingungen ein TO-NORMAL-Ereignis generiert wird:

- das Flag TO-NORMAL muss in der Eigenschaft Event_Enable aktiviert sein und
- der Present_Value darf keinem in der Eigenschaft Life_Safety_Alarm_Values enthaltenen Wert entsprechen und
- der Present_Value darf keinem in der Eigenschaft Alarm_Values enthaltenen Wert entsprechen und
- der Present_Value darf keinem in der Eigenschaft Fault_Values enthaltenen Wert entsprechen und
- der Present_Value muss für den in der Eigenschaft Time_Delay festgelegten Mindestzeitraum mit dem Wert für Alarm_Value übereinstimmen.

Neue Ereignisse werden bei einer Änderung von Mode generiert, falls das Flag TO-NORMAL in der Eigenschaft Event_Enable gesetzt ist.

7.45 Limit_Enable

Datentyp BACnetLimitEnable

Datentyp IEC BACnetLimitEnable

In Objekt Accumulator, Analog-Value, Pulse-Converter

Beschreibung Diese Eigenschaft des Typs BACnetLimitEnable enthält zwei Flags, welche die Meldung für High_Limit und Low_Limit OFFNORMAL-Ereignisse und deren Rückkehr auf Normal aktivieren bzw. deaktivieren. Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt.

7.46 Limit_Monitoring_Interval

Datentyp Unsigned

Datentyp IEC WORD

In Objekt Accumulator

Beschreibung Diese Eigenschaft des Typs Unsigned gibt den Überwachungszeitraum für die Bestimmung des Wertes von Pulse_Rate in Sekunden an. Die Verwendung eines festen oder veränderlichen Zeitfensters für die Bestimmung der Pulse Rate ist lokal unterschiedlich. Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt.

7.47 List_Of_Group_Members

Datentyp List of ReadAccessSpecification

Datentyp IEC ARRAY [0...3] of ReadAccessSpecification

In Objekt Group

Beschreibung Diese Eigenschaft ist eine Liste mit einer oder mehreren Zugriffsspezifikationen, welche die Gruppenmitglieder angeben, auf die bei Angabe des Objektes in einer Protokollaktion verwiesen wird. Jede Spezifikation für den Lesezugriff besteht aus zwei Teilen: 1) einem Object_Identifier und 2) einer „List Of Property References“. Alle Gruppenmitglieder (Objekte) müssen im selben Gerät wie das Group-Objekt sein.

Das Verketteten von Group-Objekten ist nicht zulässig, d. h. List_Of_Group_Members darf nicht auf die Eigenschaft Present_Value eines Group-Objektes verweisen.

7.48 List_Of_Object_Property_References

Datentyp	List of BACnetDeviceObjectPropertyReference
Datentyp IEC	ARRAY [0...3] of BACnetDeviceObjectPropertyReference
In Objekt	Schedule
Beschreibung	Diese Eigenschaft gibt Geräte-Identifizier, Objekt-Identifizier und Eigenschafts-Identifizier der Eigenschaften an, die zu bestimmten Zeiten an bestimmten Tagen mit bestimmten Werten geschrieben werden.

Falls diese Eigenschaft schreibbar ist, kann sie auf die Unterstützung von Verweisen auf interne Objekte des Gerätes mit dem Schedule-Objekt beschränkt werden. Ist die Eigenschaft auf Verweise zu Objekten beschränkt, die im Gerät enthalten sind, führt der Versuch, einen Verweis auf ein Objekt außerhalb des Gerätes in diese Eigenschaft zu schreiben, zur Ausgabe von Result(-) mit der Fehlerklasse PROPERTY und dem Fehlercode OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED.

Verweist diese Eigenschaft auf ein Objekt außerhalb des Gerätes, welches das Schedule-Objekt enthält, ist die Methode für das Schreiben des Wertes der Eigenschaft, die über den Verweis gesteuert wird, lokal unterschiedlich.

Die einzige Einschränkung bei der Methode zum Schreiben in diese Eigenschaft ist, dass das für das Scheduling verwendete Gerät WriteProperty unterstützen muss, um zu allen BACnet-Geräten kompatibel zu sein.

7.49 Log_Buffer

Datentyp	List of BACnetLogRecord
Datentyp IEC	ARRAY [0...3] of BACnetLogRecord
In Objekt	Trend-Log
Beschreibung	Diese Eigenschaft ist eine Liste mit einer der Buffer_Size entsprechenden Höchstanzahl an Einträgen mit Zeitstempel des Datentyps BACnetLogRecord. Jeder Eintrag enthält einen aufgezeichneten Datenwert, einen Fehler einer Datensammlung oder Zustandsänderungen des Trend-Log-Objektes. Jeder Eintrag verfügt über folgende Datenfelder:
Timestamp	Lokales Datum und Zeit bei der Erstellung des Eintrages.
LogDatum	Der ausgelesene Wert des überwachten Objektes/ der überwachten Eigenschaft, ein Fehler beim Lesen eines Wertes oder eine Zustandsänderung oder Aktion des Trend-Log-Objektes selbst.
StatusFlags	Die Eigenschaft Status_Flags des überwachten Objektes, falls vorhanden, und verfügbare zugehörige Daten des LogDatum-Werts. Ist die Eigenschaft Status_Flags nicht vorhanden, nicht verfügbar oder nicht atomar mit dem Datenwert verbunden, wird dieses Element nicht in den Protokolleintrag aufgenommen.
Verfügbare Werte für LogDatum:	
log-status	Kennzeichnet eine Zustandsänderung oder Aktion des Trend-Log-Objektes. Tritt ein durch nachfolgende Flags angegebenes Ereignis auf, wird außer bei den angegebenen Ausnahmen ein Eintrag in den Speicher geschrieben.
- log-disabled	Dieses Flag wird gesetzt, wenn das Trend-Log-Objekt deaktiviert ist, beispielsweise wenn Log_Enable „FALSE“ ist. Beginnt das Trend-Log-Objekt mit einer Aktion, wird ein Wechsel dieses Flag von „TRUE“ auf „FALSE“ angenommen und es erfolgt ein Protokolleintrag.
- buffer-purged	Dieses Flag wird auf „TRUE“ gesetzt, wenn der Speicher durch Schreiben des Wertes Null in die Eigenschaft Record_Count gelöscht wird. Nach Speichern dieses Werts wird der unmittelbar folgende Wechsel auf „FALSE“ nicht gespeichert.
boolean- valuereal- valueenum- valueunsigned- valuesigned- valuebitstring- valuenull-value	Diese Möglichkeiten geben Datenwerte an, die aus dem überwachten Objekt und der Eigenschaft ausgelesen wurden.

failure	Gibt einen Fehler beim Versuch an, einen Datenwert aus dem überwachten Objekt auszulesen. Wird der Fehler durch eine Fehlermeldung eines entfernten Gerätes angezeigt, werden Fehlerklasse und Fehlercode der Antwort gespeichert.
time-change	Gibt eine Änderung der Uhrzeiteinstellungen des Gerätes an und speichert die Zeit in Sekunden, welche die Uhr geändert wurde. Ist diese Zeit nicht bekannt, beispielsweise bei der ersten Initialisierung der Uhr, wird der Wert „Null“ eingetragen.
any-value	Gibt Datenwerte an, die aus überwachten Objekten und Eigenschaften ausgelesen werden.

Jedem Eintrag ist eine Record Number zugewiesen, deren Wert dem Total_Record_Count zum Zeitpunkt des Hinzufügens des Eintrages zum Log Buffer nach Anpassen des Record Count entspricht. Alle Clients müssen ein Zurücksetzen des Trend-Log auf den Wert „Null“ des Total_Record_Count und das Zurücksetzen des Total_Record_Count auf „1“ korrekt unterstützen.

Auf den Speicher kann nicht über das Netzwerk zugegriffen werden, außer mit dem Dienst ReadRange, um Probleme mit der segmentierten Speicherung zu vermeiden, falls eine Segmentierung erforderlich ist.

7.50 Log_DeviceObjectProperty

Datentyp BACnetDeviceObjectPropertyReference

Datentyp IEC BACnetDeviceObjectPropertyReference

In Objekt Trend-Log

Beschreibung Diese Eigenschaft des Typs BACnetDeviceObjectPropertyReference gibt „Device Identifier“, „Object Identifier“ und „Property Identifier“ der zu protokollierenden Eigenschaft an.

Falls diese Eigenschaft schreibbar ist, kann sie auf Verweise auf interne Objekte des Gerätes mit dem Trend-Log-Objekt beschränkt werden.

7.51 Log_Enable

Datentyp BOOLEAN

Datentyp IEC BOOL

In Objekt Trend-Log

Beschreibung Diese Eigenschaft des Typs BOOLEAN gibt an und bestimmt, ob die Protokollierung aktiviert ist („TRUE“) oder nicht („FALSE“). Der Wert „FALSE“ hebt das durch Start_Time und Stop_Time angegebene Intervall auf. Ist die Protokollierung andernfalls mithilfe der Eigenschaften Start_Time und Stop_Time aktiviert, werden Änderungen des Wertes der Eigenschaft Log_Enable in das Protokoll aufgenommen. Beginnt das Gerät den Betrieb, wird der Wert „TRUE“ im Protokoll gespeichert.

7.52 Log_Interval

Datentyp Unsigned

Datentyp IEC WORD

In Objekt Trend-Log

Beschreibung Diese Eigenschaft des Typs Unsigned gibt das Intervall für die Protokollierung der referenzierten Eigenschaft in Hundertstel-sekunden an. Hat diese Eigenschaft den Wert „Null“, gibt der Trend-Log COV-Benachrichtigungen für die referenzierte Eigenschaft aus. Der Wert dieser Eigenschaft darf nicht Null sein, wenn das COV_Resubscription_Interval nicht vorhanden ist. Falls diese Eigenschaft vorhanden ist, muss sie schreibbar sein.

7.53 Low_Limit

Datentyp	siehe Objekt ¹
Datentyp IEC	siehe Objekt ²
In Objekt	Accumulator (Unsigned ¹ , WORD ²), Analog Value/Pulse-Converter (REAL ¹ , REAL ²)
Beschreibung	In Objekt Accumulator: Diese Eigenschaft des Typs Unsigned gibt den Grenzwert an, den Pulse_Rate vor Generierung eines Ereignisses unterschreiten muss. Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt. In Objekt Analog-Value und Pulse-Converter: Diese Eigenschaft des Typs REAL gibt den Grenzwert an, den der Present_Value vor Generierung eines Ereignisses unterschreiten muss. Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt.

7.54 Maintenance_Required

Datentyp	siehe Objekt ¹
Datentyp IEC	siehe Objekt ²
In Objekt	Life-Safety-Point (BACnetMaintenance1, BACnetMaintenance ²), Life-Safety-Zone (BOOLEAN ¹ , BOOL ²)
Beschreibung	In Objekt Life-Safety-Point: Diese optionale Eigenschaft des Typs BACnetMaintenance gibt die für den Life-Safety-Point erforderliche Wartung an. Dies kann eine regelmäßige Wartung oder eine parameterbestimmte Wartung sein, z. B. der Verschmutzungswert eines entsprechenden Detektors. Dies wird lokal bestimmt. In Objekt Life-Safety-Zone: Diese optionale Eigenschaft des Typs BOOLEAN gibt an, dass eine Wartung eines oder mehrerer Life-Safety-Points erforderlich ist, die Mitglied dieser Zone sind.

7.55 Max_Pres_Value

Datentyp Unsigned

Datentyp IEC WORD

In Objekt Accumulator

Beschreibung Diese Eigenschaft des Typs Unsigned gibt den Höchstwert für die Eigenschaft Present_Value an.

7.56 Maximum_Value

Datentyp REAL

Datentyp IEC REAL

In Objekt Averaging

Beschreibung Diese Eigenschaft des Typs REAL gibt den höchsten im Speicherfenster enthaltenen Wert für die aktuellsten Window_Samples oder die aktuelle Anzahl an Samples (Valid_Samples) an, falls weniger Samples als Window_Samples erfasst wurden. Nach einem Neustart des Gerätes oder nachdem Attempted_Samples, Object_Property_Reference, Window_Samples oder Window_Interval mit BACnet-Diensten geschrieben wurden, hat Maximum_Value den Wert **-INF**.

7.57 Maximum_Value_Timestamp

Datentyp BACnetDateTime

Datentyp IEC BACnetDateTime

In Objekt Averaging

Beschreibung Diese optionale Eigenschaft des Typs BACnetDateTime gibt an, wann der Wert von Maximum_Value erfasst wurde.

7.58 Member_Of

Datentyp	List of BACnetDeviceObjectReference
Datentyp IEC	ARRAY [0...3] of BACnetDeviceObjectReference
In Objekt	Life-Safety-Point, Life-Safety-Zone
Beschreibung	Diese optionale Eigenschaft des Typs „List of BACnetDeviceObjectReference“ gibt die Life-Safety-Point-/Life-Safety-Zone-Objekte an, für die dieses Life-Safety-Zone-Objekt als Mitglied gilt. Jedes Objekt der Liste Member_Of muss ein Life-Safety-Zone-Objekt sein. Diese Eigenschaft kann auf die Unterstützung von Verweisen auf interne Objekte des Gerätes mit dem Life-Safety-Point-/Life-Safety-Zone-Objekt beschränkt werden. Ist die Eigenschaft schreibbar und auf Verweise zu Objekten beschränkt, die im Gerät enthalten sind, führt der Versuch, einen Verweis auf ein Objekt außerhalb des Gerätes in diese Eigenschaft zu schreiben, zur Ausgabe von Result(-) mit der Fehlerklasse PROPERTY und dem Fehlercode OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED.

7.59 Minimum_Off_Time

Datentyp	Unsigned32
Datentyp IEC	DWORD
In Objekt	Binary-Value
Beschreibung	Diese Eigenschaft des Typs Unsigned32 gibt den Mindestwert in Sekunden an, den der Present_Value nach dem Schreiben eines Wertes in die Eigenschaft Present_Value im Zustand INACTIVE verbleibt, wenn die Eigenschaft beim Schreiben den Zustand INACTIVE annimmt.

7.60 Minimum_On_Time

Datentyp Unsigned32

Datentyp IEC DWORD

In Objekt Binary-Value

Beschreibung Diese Eigenschaft des Typs Unsigned32 gibt den Mindestwert in Sekunden an, den der Present_Value nach dem Schreiben eines Wertes in die Eigenschaft Present_Value im Zustand ACTIVE verbleibt, wenn die Eigenschaft beim Schreiben den Zustand ACTIVE annimmt.

7.61 Minimum_Value

Datentyp REAL

Datentyp IEC REAL

In Objekt Averaging

Beschreibung Diese Eigenschaft des Typs REAL gibt den niedrigsten Wert des Speicherfensters für die aktuellsten Window_Samples oder die aktuelle Sample-Anzahl (Valid_Samples) an, falls weniger Samples als Window_Samples vorhanden sind. Nach einem Neustart des Gerätes oder nach dem Schreiben von Attempted_Samples, Object_Property_Reference, Window_Samples oder Window_Interval mit BACnet-Diensten hat Minimum_Value den Wert INF.

7.62 Minimum_Value_Timestamp

Datentyp BACnetDateTime

Datentyp IEC BACnetDateTime

In Objekt Averaging

Beschreibung Diese optionale Eigenschaft des Typs BACnetDateTime gibt an, wann der Wert von Minimum_Value erfasst wurde.

7.63 Mode

Datentyp	siehe Objekt ¹
Datentyp IEC	siehe Objekt ²
In Objekt	Life-Safety-Point/Life-Safety-Zone (BACnetLifeSafetyMode ¹ , BACnetLifeSafetyMode ²)
Beschreibung	Diese schreibbare Eigenschaft des Typs BACnetLifeSafetyMode gibt die gewünschte Betriebsart für das Life-Safety-Point-Objekt an. Das Life-Safety-Point-Objekt generiert CHANGE_OF_LIFE_SAFETY-Ereignisbenachrichtigungen für jeden Mode-Wechsel, falls die entsprechenden Flags TO-OFFNORMAL, TO-FAULT oder TO-NORMAL in der Eigenschaft Event_Enable gesetzt sind, siehe Bedingungen für die BACnet-Eigenschaften „Life_Safety_Alarm_Values“, „Fault_Values“ und „Alarm_Values“.

7.64 Modification_Date

Datentyp	BACnetDateTime
Datentyp IEC	BACnetDateTime
In Objekt	File
Beschreibung	Diese Eigenschaft des Typs BACnetDateTime gibt an, wann das Objekt zum letzten Mal geändert wurde. Ein File-Objekt gilt als verändert, wenn es erstellt wird oder wenn etwas in das Objekt geschrieben wird.

7.65 Notification_Class

Datentyp Unsigned

Datentyp IEC WORD

In Objekt Accumulator, Analog-Value, Binary-Value, Life-Safety-Point, Life-Safety-Zone, Multi-State-Value, Notification-Class, Pulse-Converter, Trend-Log

Beschreibung **In Objekt Accumulator, Analog-Value, Binary-Value, Life-Safety-Point, Life-Safety-Zone, Multi-State-Value, Pulse-Converter und Trend-Log:**

Diese Eigenschaft des Typs Unsigned gibt die bei der Handhabung und Generierung von Ereignisbenachrichtigungen für dieses Objekt zu verwendende Meldeklasse an. Die Eigenschaft Notification_Class verweist eindeutig auf ein Notification-Class-Objekt, dessen Eigenschaft Notification_Class denselben Wert aufweist. Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt (außer bei dem Live-Safety-Zone-Objekt).

In Objekt Notification-Class:

Diese Eigenschaft des Typs Unsigned gibt den numerischen Wert dieser Notification-Class an und muss der Instanznummer des Notification-Class-Objektes entsprechen. Ereignisauslösende Objekte verwenden diese Nummer für indirekte Verweise auf das Notification-Class-Objekt.

7.66 Notification_Threshold

Datentyp Unsigned32

Datentyp IEC DWORD

In Objekt Trend-Log

Beschreibung Diese Eigenschaft des Typs Unsigned32 gibt den Wert von Records_Since_Notification an, bei dem die Benachrichtigung erfolgt. Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt.

7.67 Notify_Type

Datentyp	BACnetNotifyType
Datentyp IEC	BACnetNotifyType
In Objekt	Accumulator, Analog-Value, Binary-Value, Life-Safety-Point, Life-Safety-Zone, Multi-State-Value, Pulse-Converter, Trend-Log
Beschreibung	Diese Eigenschaft des Typs BACnetNotifyType gibt an, ob die von diesem Objekt generierten Benachrichtigungen Ereignisse oder Alarmer sind. Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt. Für das Life-Safety-Zone-Objekt ist diese Eigenschaft nicht erforderlich, wenn das Objekt objektinternes Melden unterstützt.

7.68 Number_Of_States

Datentyp	Unsigned
Datentyp IEC	WORD
In Objekt	Multi-State-Value
Beschreibung	Diese Eigenschaft des Typs Unsigned gibt die Anzahl der Zustände für die Eigenschaft Present_Value an. Die Eigenschaft Number_Of_States muss immer einen Wert größer als Null haben. Bei Änderung des Wertes dieser Eigenschaft wird die Größe des Feldes State_Text falls vorhanden auf denselben Wert geändert.

7.69 Object_Identifier

Datentyp	BACnetObjectIdentifier
Datentyp IEC	BACnetObjectIdentifier
In Objekt	Accumulator, Analog-Value, Averaging, Binary-Value, Calendar, Command, File, Group, Life-Safety-Point, Life-Safety-Zone, Multi-State-Value, Notification-Class, Pulse-Converter, Schedule, Trend-Log
Beschreibung	Diese Eigenschaft des Typs BACnetObjectIdentifier ist ein numerischer Code für die Bestimmung des Objektes. Er muss im enthaltenden BACnet-Gerät eindeutig sein.

7.70 Object_Name

Datentyp	CharacterString
Datentyp IEC	STRING
In Objekt	Accumulator, Analog-Value, Averaging, Binary-Value, Calendar, Command, File, Group, Life-Safety-Point, Life-Safety-Zone, Multi-State-Value, Notification-Class, Pulse-Converter, Schedule, Trend-Log
Beschreibung	Diese Eigenschaft des Typs CharacterString gibt den Namen des Objektes an, der innerhalb des verwaltenden BACnet-Gerätes eindeutig ist. Die Mindestlänge für dieser Zeichenkette beträgt ein Zeichen. Der für Object_Name verwendete Zeichensatz darf ausschließlich druckbare Zeichen enthalten.

7.71 Object_Property_Reference

Datentyp BACnetDeviceObjectPropertyReference

Datentyp IEC BACnetDeviceObjectPropertyReference

In Objekt Averaging

Beschreibung Diese Eigenschaft des Typs BACnetDeviceObjectPropertyReference gibt Objekt und Eigenschaft für den Wert an, der in Window_Interval erfasst wird. Das referenzierte Objekt kann sich im Gerät befinden, welches das Averaging-Objekt enthält, optional kann das Averaging-Objekt Verweise auf Objekteigenschaften in anderen Geräten unterstützen. Externe Verweise können auf bestimmte BACnet-Geräte beschränkt werden. Die referenzierte Objekteigenschaft muss einen der numerischen Datentypen BOOLEAN, INTEGER, Unsigned, Enumerated oder REAL besitzen. Alle erfassten Daten werden zur Berechnung in REAL konvertiert.

Der boolesche Wert „FALSE“ wird als „Null“ gewertet, „TRUE“ als „1“. Enumerierte Datentypen werden als Unsigned-Werte behandelt. Unterstützt eine Implementierung das Schreiben in Object_Property_Reference, werden beim Schreiben von Object_Property_Reference mit BACnet-Diensten alle Samples im Speicher ungültig, Attempted_Samples erhält den Wert „Null“, Valid_Samples erhält den Wert „Null“, Minimum_Value erhält den Wert „INF“, Average_Value erhält den Wert „NaN“ und Maximum_Value den Wert „-INF“.

7.72 Object_Type

Datentyp	BACnetObjectType
Datentyp IEC	BACnetObjectType
In Objekt	Accumulator, Analog-Value, Averaging, Binary-Value, Calendar, Command, File, Group, Life-Safety-Point, Life-Safety-Zone, Multi-State-Value, Notification-Class, Pulse-Converter, Schedule, Trend-Log
Beschreibung	Diese Eigenschaft des Typs BACnetObjectType gibt die Zugehörigkeit zu einer bestimmten Objektklasse an. Der Wert dieser Eigenschaft kann ACCUMULATOR, ANALOG_VALUE, AVERAGING, BINARY_VALUE, CALENDAR, COMMAND, EVENT_ENROLLMENT, FILE, GROUP, LIFE_SAFETY_POINT, LIFE_SAFETY_ZONE, MULTISTATE_INPUT, MULTISTATE_OUTPUT, MULTISTATE_VALUE, NOTIFICATION_CLASS, PULSE_CONVERTER, SCHEDULE oder TREND_LOG sein.

7.73 Operation_Expected

Datentyp	BACnetLifeSafetyOperation
Datentyp IEC	BACnetLifeSafetyOperation
In Objekt	Life-Safety-Point, Life-Safety-Zone
Beschreibung	Die Eigenschaft Operation_Expected des Typs BACnetLifeSafetyOperation gibt die nächste von diesem Objekt erwartete Aktion für die Handhabung bestimmter Life-Safety-Situationen an.

7.74 Out_Of_Service

Datentyp BOOLEAN

Datentyp IEC BOOL

In Objekt Accumulator, Analog-Value, Binary-Value, Life-Safety-Point, Life-Safety-Zone, Multi-State-Value, Pulse-Converter, Schedule

Beschreibung **In Objekt Accumulator:**

Die Eigenschaft Out_Of_Service des Typs BOOLEAN zeigt an, ob der physikalische Eingang, den das Objekt darstellt, aktiv ist („TRUE“) oder nicht („FALSE“). Wenn der Wert von Out_Of_Service „TRUE“ ist, sind die Eigenschaften Present_Value und Pulse_Rate nicht an den physikalischen Eingang gekoppelt und Veränderungen an diesem Eingang werden nicht verfolgt. Weiterhin müssen die Eigenschaft Reliability und der entsprechende Zustand der FAULT-Flag der Eigenschaft Status_Flags vom physikalischen Eingang getrennt sein, wenn Out_Of_Service „TRUE“ ist. Wenn die Eigenschaft Out_Of_Service „TRUE“ ist, können die Eigenschaften Present_Value, Pulse_Rate und Reliability zur Simulation bestimmter Zustände oder für Prüfzwecke auf einen beliebigen Wert gesetzt werden. Weitere Funktionen, die vom Zustand der Eigenschaften Present_Value, Pulse_Rate oder Reliability abhängen, müssen auf Werteänderungen reagieren, als ob die Änderungen am physikalischen Eingang vorliegen, wenn Out_Of_Service „TRUE“ ist.

In Objekt Analog-Value und Binary-Value:

Die Eigenschaft Out-Off-Service des Typs BOOLEAN gibt an, ob der Present_Value des Analog-Value-Objektes und des Binary-Value-Objektes mittels lokaler Software des BACnet-Gerätes, in dem sich das Objekt befindet, verändert werden kann („TRUE“) oder nicht („FALSE“). Wenn Out_Of_Service „TRUE“ ist, kann in die Eigenschaft Present_Value geschrieben werden. Sind die Eigenschaften Priority_Array und Relinquish_Default vorhanden, wird das Schreiben der Eigenschaft Present_Value über den BACnet-Command-Prioritätsmechanismus gesteuert.

In Objekt Life-Safety-Point:

Die Eigenschaft Out_Of_Service des Typs BOOLEAN zeigt an, ob Eingänge oder Prozesse, die das Objekt darstellen, aktiv sind („TRUE“) oder nicht („FALSE“). Wenn der Wert von Out_Of_Service „TRUE“ ist, ist die Eigenschaft Present_Value nicht an Eingänge oder Prozesse gekoppelt und Veränderungen werden nicht verfolgt. Weiterhin müssen die Eigenschaft Reliability und der entsprechende Zustand der FAULT-Flag der Eigenschaft Status_Flags vom physikalischen Eingang getrennt sein, wenn Out_Of_Service „TRUE“ ist. Wenn die Eigenschaft Out_Of_Service „TRUE“ ist, können die Eigenschaften Present_Value und Reliability zur Simulation bestimmter Zustände oder für Prüfzwecke auf einen beliebigen Wert gesetzt werden. Weitere Funktionen, die vom Zustand der Eigenschaften Present_Value oder Reliability abhängen, müssen auf Werteänderungen reagieren, als ob die Änderungen an den Eingängen oder Prozessen vorliegen, wenn Out_Of_Service „TRUE“ ist.

In Objekt Multi-State-Value:

Die Eigenschaft Out_Of_Service des Typs BOOLEAN zeigt an, ob die Eigenschaft Present_Value des Multi-State-Value-Objektes durch lokale Software des BACnet-Gerätes, in dem sich das Objekt befindet, verändert werden kann („TRUE“) oder nicht („FALSE“). Wenn Out_Of_Service „TRUE“ ist, kann in die Eigenschaft Present_Value geschrieben werden. Außerdem müssen dann die Eigenschaft Reliability und der entsprechende Zustand des FAULT-Flags der Eigenschaft Status_Flags getrennt sein. Ist Out_Of_Service „TRUE“, können die Eigenschaften Present_Value und Reliability zur Simulation bestimmter Zustände oder für Prüfzwecke auf einen beliebigen Wert gesetzt werden. Weitere Funktionen, die vom Zustand der Eigenschaften Present_Value oder Reliability abhängen, müssen auf Änderungen dieser Eigenschaften reagieren, wenn Out_Of_Service „TRUE“ ist. Sind die Eigenschaften Priority_Array und Relinquish_Default vorhanden, wird das Schreiben in die Eigenschaft Present_Value nach wie vor über den BACnet-Command-Prioritätsmechanismus gesteuert.

In Objekt Pulse-Converter:

Die Eigenschaft Out_Of_Service des Typs BOOLEAN zeigt an, ob der Eingang, den das Objekt direkt darstellt, aktiv ist („TRUE“) oder nicht („FALSE“). „Direkt“ bedeutet hier, dass die Eigenschaft Input_Reference in diesem Objekt nicht vorhanden ist. Wenn der Wert von Out_Of_Service „TRUE“ ist, ist die Eigenschaft Present_Value nicht an die Eigenschaft Count gekoppelt und Veränderungen an diesem Eingang werden nicht verfolgt. Weiterhin müssen die Eigenschaft Reliability und der entsprechende Zustand des FAULT-Flags der Eigenschaft Status_Flags vom Eingang getrennt sein, wenn Out_Of_Service „TRUE“ ist. Wenn die Eigenschaft Out_Of_Service „TRUE“ ist, können die Eigenschaften Present_Value und Reliability zur Simulation bestimmter Zustände oder für Prüfzwecke auf einen beliebigen Wert gesetzt werden. Weitere Funktionen, die vom Zustand der Eigenschaften Present_Value oder Reliability abhängen, müssen auf Änderungen der Eigenschaften reagieren, als ob die Änderungen am Eingang vorliegen, wenn Out_Of_Service „TRUE“ ist.

Ist die Eigenschaft Input_Reference vorhanden, wird der Zustand der Eigenschaft Out_Of_Service des durch Input_Reference referenzierten Objektes nicht durch die Eigenschaft Out_Of_Service des Pulse-Converter-Objektes angezeigt.

In Objekt Schedule:

Die Eigenschaft Out_Of_Service des Typs BOOLEAN gibt an, ob die internen Berechnungen des Schedule-Objektes zur Bestimmung des Wertes der Eigenschaft Present_Value verwendet wird („TRUE“) oder nicht („FALSE“). Wenn der Wert von Out_Of_Service „TRUE“ ist, ist die Eigenschaft Present_Value nicht an die internen Berechnungen gekoppelt und Veränderungen anderer Eigenschaften werden nicht verfolgt. Weitere Funktionen, die vom Zustand der Eigenschaft Present_Value abhängen, beispielsweise das Schreiben an die Mitglieder der List_Of_Object_Property_References, müssen auf Änderungen dieser Eigenschaft reagieren, als ob die Änderungen durch interne Berechnungen erfolgen, wenn Out_Of_Service „TRUE“ ist.

7.75 Prescale

Datentyp BACnetPrescale

Datentyp IEC BACnetPrescale

In Objekt Accumulator

Beschreibung Diese Eigenschaft des Typs BACnetPrescale gibt die Koeffizienten für die Konvertierung der von Messgeräten generierten Impulssignale in den durch Present_Value angezeigten Wert an. Die Konvertierung erfolgt mittels Integer-Arithmetik, so dass dabei keine Messsignale verloren gehen. Die Koeffizienten stellen unter Umständen eine vor Empfang der Eingangsimpulse durch das Accumulator-Objekt durchgeführte Konvertierung dar oder sie werden vom Accumulator-Objekt zur Konvertierung von Eingangsimpulsen in den durch Present_Value angezeigten Wert verwendet. Dies ist lokal unterschiedlich.

Folgende Koeffizienten sind vorhanden:

multiplier Zähler des als Integer-Verhältnis ausgedrückten Konvertierungsfaktors

moduloDivide Nenner des als Integer-Verhältnis ausgedrückten Konvertierungsfaktors

Der Konvertierungsalgorithmus wird wie folgt mit einer nicht angezeigten Variable „Accumulator“ durchgeführt:

Für jeden Eingangsimpuls:

Den Wert von multiplier einem Accumulator hinzufügen.

Ist der Accumulator größer oder gleich dem Wert von moduloDivide, dann den Wert von Present_Value um „1“ erhöhen und den Wert des Accumulator um den Wert von moduloDivide verringern.

Dieser Vorgang unterstützt nicht integrale Verhältnisse von Messimpulsen für Present_Value. Bei einer elektrischen Messanwendung ist die Ausgabe der Spannungs- und Strommessung beispielsweise 9000/1200 (Maßstab/Spannung*Strom) Impulse pro kWh, sodass das Accumulator-Objekt 2/15 kWh/Impuls erfassen muss. Mit diesem Algorithmus können derartige Impulse genau erfasst und angezeigt werden, wenn die Einheit für Present_Value KILOWATT_HOURS ist.

7.76 Present_Value

Datentyp Unsigned

Datentyp IEC WORD

In Objekt Accumulator, Multi-State-Value, Command

Beschreibung **In Objekt Accumulator:**
Diese Eigenschaft des Typs Unsigned gibt die Anzahl der Eingangsimpulse an, die seit dem letzten Setzen des Wertes durch Schreiben in die Eigenschaft Value_Set erfasst wurde. Die Werte sind vorkaliert, wenn die Eigenschaft Prescale vorhanden ist.
Der Wert dieser Eigenschaft muss im Bereich Null bis Max_Pres_Value sein. Alle Aktionen der Eigenschaft Present_Value werden „modulo“ (Max_Pres_Value+1) ausgeführt.
Diese Eigenschaft muss schreibbar sein, wenn Out_Of_Service „TRUE“ ist.

In Objekt Multi-State-Value:
Diese Eigenschaft des Typs Unsigned gibt den logischen Zustand eines Multi-State-Value an. Der logische Zustand des Multi-State-Value ist einer von „n“ Zuständen, wobei „n“ ein in der Eigenschaft Number_Of_States definierter Zustand ist. Die Methode für die Ermittlung des Present_Value ist lokal unterschiedlich. Die Eigenschaft Present_Value muss immer einen Wert größer als „Null“ haben. Present_Value kann optional Befehle empfangen. Kann Present_Value für eine bestimmte Objektinstanz Befehle empfangen, müssen die Eigenschaften Priority_Array und Relinquish_Default ebenfalls für diese Instanz vorhanden sein. Die Eigenschaft Present_Value muss schreibbar sein, wenn Out_Of_Service „TRUE“ ist.

In Objekt Command:

Diese Eigenschaft des Typs Unsigned gibt an, welche Aktionen das Command-Objekt ausführt oder bereits ausgeführt hat. Beim Schreiben in die Eigenschaft Present_Value des Command-Objektes löst diese Eigenschaft bestimmte Aktionen des Command-Objektes aus, welche die Werte anderer Objekteigenschaften verändern. In den Present_Value kann ein beliebiger Wert von „0“ bis zur maximalen Anzahl von Aktionen, welche die Eigenschaft Action unterstützt, geschrieben werden. Ist der Present_Value geschrieben, beginnt das Command-Objekt mit der Ausführung bestimmter Aktionen. Der neue Wert der Eigenschaft Present_Value bestimmt, welche Aktionssequenz das Command-Objekt ausführt. Die Aktionen sind in der Eigenschaft Action angegeben, die ein Feld mit Listen auszuführender Aktionen ist. Das Feld wird mit dem geschriebenen Wert indiziert. Eine bestimmte Liste kann leer sein, in diesem Fall wird keine Aktion ausgeführt. Ist die Liste nicht leer, schreibt das Command-Objekt für jede Aktion in der Liste einen bestimmten Wert in eine bestimmte Eigenschaft eines bestimmten Objektes in einem bestimmten BACnet-Gerät.

Datentyp	List of ReadAccessResult
Datentyp IEC	ARRAY [0...3] of ReadAccessResult;
In Objekt	Group
Beschreibung	Diese Eigenschaft ist eine Liste mit den Werten aller in der List_Of_Group_Members angegebenen Eigenschaften. Die Eigenschaft hat nur Lesezugriff, sie kann nicht für das Schreiben von Werten in Gruppenmitglieder verwendet werden. Die Liste Present_Value wird bei jedem Lesen der Eigenschaft durch Erfassen der Eigenschaften der Listenmitglieder aktualisiert. Dies erfolgt, um mögliche Probleme durch unterschiedliche Werte bei einem Speichern der Liste zu vermeiden.

Datentyp	BACnetLifeSafetyState
Datentyp IEC	BACnetLifeSafetyState
In Objekt	Life-Safety-Point, Life-Safety-Zone
Beschreibung	Diese Eigenschaft des Typs BACnetLifeSafetyState gibt den Zustand des Life-Safety-Point-/Life-Safety-Zone-Objektes an. Die Methode für die Ermittlung von Present_Value ist lokal unterschiedlich. Present_Value kann bis zum Zurücksetzen NON-NORMAL-Zustände annehmen. Die Eigenschaft Present_Value muss schreibbar sein, wenn Out_Of_Service „TRUE“ ist.

Datentyp	REAL
Datentyp IEC	REAL
In Objekt	Analog-Value, Pulse-Converter
Beschreibung	In Objekt Analog-Value: Diese Eigenschaft des Typs REAL gibt den aktuellen Wert von Analog-Value in Engineering-Einheiten an. Present_Value kann optional Befehle empfangen. Kann Present_Value für eine bestimmte Objektinstanz Befehle empfangen, müssen die Eigenschaften Priority_Array und Relinquish_Default ebenfalls für diese Instanz vorhanden sein. Die Eigenschaft Present_Value muss schreibbar sein, wenn Out_Of_Service „TRUE“ ist. In Objekt Pulse-Converter: Diese Eigenschaft des Typs REAL gibt den akkumulierten Wert des zu messenden Einganges an. Dieser wird durch multiplizieren des aktuellen Wertes der Eigenschaft Count mit dem Wert der Eigenschaft Scale_Factor berechnet. Der Wert von Present Value kann durch Schreiben in die Eigenschaft Adjust_Value angepasst werden. Die Eigenschaft Present_Value muss schreibbar sein, wenn Out_Of_Service „TRUE“ ist.

Datentyp	beliebig
Datentyp IEC	ARRAY [0...10] of BYTE
In Objekt	Schedule
Beschreibung	Diese Eigenschaft gibt den aktuellen Wert des Zeitplanes an. Dies kann jeder einfache Datentyp sein. Aus diesem Grund kann der Zeitplan für die meisten analogen, binären und enumerierten Werte verwendet werden. Diese Eigenschaft muss schreibbar sein, wenn Out_Of_Service „TRUE“ ist (siehe Kapitel „Out_Of_Service“). Jede Änderung dieser Eigenschaft muss an alle Mitglieder der List_Of_Object_Property_References geschrieben werden. Bei einem Fehler bei der Übergabe an ein Listenmitglied fährt das Schedule-Objekt mit der Übergabe an die verbleibenden Mitglieder fort. Die standardmäßige Berechnung des Wertes für die Eigenschaft Present_Value läuft wie nachfolgend dargestellt ab (der tatsächlich verwendete Algorithmus ist lokal unterschiedlich, muss aber zu denselben Ergebnissen wie dieser führen): 1. Ermitteln des Exception_Schedule-Feldelementes mit der höchsten relativen Priorität (wie in Kapitel „Exception_Schedule“ definiert), das für den aktuellen Tag gilt und dessen aktueller Wert (siehe Methode unten) nicht „NULL“ ist. Diesen Wert der Eigenschaft Present_Value zuweisen. 2. Falls der Present_Value nicht im vorherigen Schritt zugewiesen wurde, den Wert des Weekly_Schedule-Feldelementes für den aktuellen Tag ermitteln. Falls dessen Wert nicht „NULL“ ist, den Wert der Eigenschaft Present_Value zuweisen. 3. Falls der Present_Value nicht in den vorherigen Schritten zugewiesen wurde, dem Present_Value den Wert der Eigenschaft Schedule_Default zuweisen.

Mit der Methode zur Ermittlung des aktuellen Wertes eines Zeitplans (Exception oder Weekly) wird das letzte Element in der Liste für BACnetTimeValues gesucht, das zur bzw. vor der aktuellen Zeit auftritt. Der Wert dieses Elementes wird als aktueller Wert für den Zeitplan verwendet. Wird kein solches Element gefunden, ist der aktuelle Wert für den Zeitplan „NULL“.

Diese Berechnungen können jederzeit ausgeführt und so der korrekte Wert für den Present_Value ermittelt werden. Die Berechnungen müssen täglich um 00:00 ausgeführt werden, wenn das Gerät zurückgesetzt wird, wenn Eigenschaften mit möglichen Auswirkungen auf die Ergebnisse geändert werden, wenn die Gerätezeit um einen Wert verändert wird, der Auswirkungen auf das Ergebnis der Berechnung haben könnte und bei Bedarf, um stets einen korrekten Wert für den Present_Value sicherzustellen.

Zu beachten ist, dass der Eigenschaft Present_Value täglich um 00:00 der Wert der Eigenschaft Schedule_Default zugewiesen wird, außer es besteht ein Eintrag für 00:00 an diesem Tag. Ist der logische Beginn eines Schedule-Ereignisses an einem Tag und endet an einem anderen Tag, muss für den zweiten Tag um 00:00 ein Eintrag im Zeitplan vorhanden sein, ebenso für weitere Tage im Verlauf des Ereignisses, um korrekte Ergebnisse bei der Berechnung des Present_Value sicherzustellen.

Datentyp	BACnetBinaryPV
Datentyp IEC	BACnetBinaryPV
In Objekt	Binary-Value
Beschreibung	Diese Eigenschaft des Typs BACnetBinaryPV gibt den logischen Zustand von Binary-Value an. Der logische Zustand ist entweder INACTIVE oder ACTIVE. Der Present_Value kann optional Befehle empfangen. Kann Present_Value für eine bestimmte Objektinstanz Befehle empfangen, müssen die Eigenschaften Priority_Array und Relinquish_Default ebenfalls für diese Instanz vorhanden sein. Die Eigenschaft Present_Value muss schreibbar sein, wenn Out_Of_Service „TRUE“ ist.

Datentyp	BOOLEAN
Datentyp IEC	BOOL
In Objekt	Calendar
Beschreibung	Diese Eigenschaft des Typs BOOLEAN gibt den aktuellen Wert des Kalenders an: „TRUE“ wenn das aktuelle Datum in der Date_List enthalten ist, „FALSE“ wenn nicht.

7.77 Priority

Datentyp	BACnetARRAY [3] of Unsigned
Datentyp IEC	ARRAY [1...3] of WORD
In Objekt	Notification-Class
Beschreibung	Diese Eigenschaft des Typs „BACnetARRAY [3] of Unsigned“ gibt die Priorität für Ereignisbenachrichtigungen für TO-OFFNORMAL-, TO-FAULT- und TO-NORMAL-Ereignisse an. Es können Prioritätswerte von „0“ bis „255“ vergeben werden. Ein niedrigerer Wert bedeutet eine höhere Priorität (siehe Tabelle in Kapitel „Prioritätsklassifizierung von Alarmen und Ereignissen“).

7.78 Priority_Array

Datentyp	BACnetPriorityArray
Datentyp IEC	ARRAY [1...16] of BOOL
In Objekt	Analog-Value, Binary-Value
Beschreibung	Diese Eigenschaft ist ein Read-Only-Feld mit priorisierten Befehlen für dieses Objekt (siehe Dokumentation des WAGO-BACnet/IP-Controllers 750-830 für eine Beschreibung des Mechanismus zur Priorisierung). Wenn die Eigenschaft Relinquish_Default oder Priority_Array vorhanden ist, müssen beide vorhanden sein. Kann Present_Value Befehle empfangen, müssen die Eigenschaften Priority_Array und Relinquish_Default vorhanden sein und Present_Value muss für das Binary-Value schreibbar sein.

Datentyp	BACnetPriorityArray
Datentyp IEC	BACnetPriorityArray
In Objekt	Multi-State-Value
Beschreibung	Diese Eigenschaft ist ein Read-Only-Feld mit priorisierten Befehlen für dieses Objekt (siehe Dokumentation des WAGO-BACnet/IP-Controllers 750-830 für eine Beschreibung des Mechanismus zur Priorisierung). Wenn die Eigenschaft Relinquish_Default oder Priority_Array vorhanden ist, müssen beide vorhanden sein. Kann der Present_Value Befehle empfangen, müssen die Eigenschaften Priority_Array und Relinquish_Default vorhanden sein.

Datentyp	Unsigned (1...16)
Datentyp IEC	BYTE
In Objekt	Schedule
Beschreibung	Diese Eigenschaft bestimmt die Priorität für Befehle der referenzierten Eigenschaften. Sie entspricht dem Parameter Priority des Dienstes WriteProperty. Der Wert ist ein Unsigned-Integer im Bereich „1“ bis „16“, wobei „1“ die höchste und „16“ die niedrigste Priorität darstellt.

7.79 Priority_For_Writing

Datentyp	Unsigned (1...16)
Datentyp IEC	BYTE
In Objekt	Schedule
Beschreibung	Diese Eigenschaft bestimmt die Priorität für Befehle der referenzierten Eigenschaften. Sie entspricht dem Parameter Priority des Dienstes WriteProperty. Der Wert ist ein Unsigned-Integer im Bereich „1“ bis „16“, wobei „1“ die höchste und „16“ die niedrigste Priorität darstellt.

7.80 Pulse_Rate

Datentyp Unsigned

Datentyp IEC WORD

In Objekt Accumulator

Beschreibung Diese Eigenschaft des Typs Unsigned gibt die Anzahl der Eingangsimpulse an, die im aktuellsten durch Limit_Monitoring_Interval angegebenen Intervall erfasst wurden. Die Methode für die Zuweisung des Eingangssignals zu dem Wert, der in dieser Eigenschaft angezeigt wird, ist lokal unterschiedlich. Diese Eigenschaft muss schreibbar sein, wenn Out_Of_Service „TRUE“ ist.

7.81 Read_Only

Datentyp BOOLEAN

Datentyp IEC BOOL

In Objekt File

Beschreibung Diese Eigenschaft des Typs BOOLEAN gibt an, ob Dateidaten durch einen BACnet-AtomicWriteFile-Dienst verändert werden dürfen („FALSE“) oder nicht („TRUE“).

7.82 Recipient_List

Datentyp List of BACnetDestination

Datentyp IEC ARRAY [0...3] of BACnetDestination

In Objekt Notification-Class

Beschreibung Diese Eigenschaft des Typs „List of BACnetDestination“ gibt eine Liste einer oder mehrerer Ziele an, an die Benachrichtigungen gesendet werden, wenn ereignisauslösende Objekte, die diese Klasse zur Bestimmung eines Ereignisses verwenden, ein Ereignis erkennen. Die Parameter der Ziele werden wie folgt definiert (siehe folgende Tabelle).

Tabelle 70: Komponenten einer BACnetDestination

Parameter	Typ	Beschreibung
Valid Days	BACnetDaysOfWeek	Wochentage, an denen dieses Ziel zwischen „From Time“ und „To Time“ verwendet werden kann.
From Time, To Time	Time	Zeitfenster, in dem das Ziel an den „Valid Days“ verwendet werden kann.
Recipient	BACnetRecipient	Zielgerät(e), die Benachrichtigungen empfangen
Process Identifier	Unsigned32	Handle eines Prozesses im empfangenden Gerät, das die Ereignisbenachrichtigung empfängt
Issue Confirmed Notifications	Boolean	„TRUE“, wenn zu bestätigende Nachrichten gesendet werden sollen, „FALSE“, wenn gesendete Benachrichtigungen keine Bestätigung erfordern
Transitions	BACnet Event Transition Bits	drei Flags, die Wechsel zwischen den Zuständen {TO-OFFNORMAL, TO-FAULT, TO-NORMAL} angeben, für die der Empfänger geeignet ist.

7.83 Record_Count

Datentyp Unsigned

Datentyp IEC IEC_DWORD

In Objekt Trend-Log

Beschreibung Diese Eigenschaft des Typs Unsigned gibt die Dateigröße in Records an. Die Eigenschaft Record_Count darf nur vorhanden sein, wenn der File_Access_Type RECORD_ACCESS verwendet wird. Falls die Anzahl der Records durch Schreiben in die Datei verändert werden kann, muss diese Eigenschaft schreibbar sein.

Das Schreiben in die Eigenschaft Record_Count mit einem geringeren als dem aktuellen Wert kürzt die Datei an der angegebenen Position. Das Schreiben eines Record_Count von „0“ löscht alle Daten der Datei, nicht jedoch das Dateiojekt selbst. Das Schreiben in die Eigenschaft Record_Count mit einem größeren als dem aktuellen Wert erweitert die Dateigröße. Der Wert der neuen Oktette der Datei ist lokal unterschiedlich.

Geräte können die für das Schreiben von Record_Count zulässigen Werte einschränken. Dies bedeutet insbesondere, dass Geräte das Löschen des Dateiinhaltes durch Schreiben des Wertes „Null“ zulassen können, nicht jedoch eine beliebige Kürzung oder Erweiterung.

Datentyp Unsigned32

Datentyp IEC DWORD

In Objekt Trend-Log

Beschreibung Diese Eigenschaft des Typs Unsigned32 gibt die Anzahl der derzeit im Protokollspeicher enthaltenen Einträge an. Schreiben von „Null“ in diese Eigenschaft löscht alle Einträge und Records_Since_Notification wird auf „Null“ zurückgesetzt. Nach der Ausführung wird dieses Ereignis als erster Eintrag in das Protokoll geschrieben.

7.84 Records_Since_Notification

Datentyp	Unsigned32
Datentyp IEC	DWORD
In Objekt	Trend-Log
Beschreibung	Diese Eigenschaft des Typs Unsigned32 gibt die seit der letzten Benachrichtigung oder, falls keine vorherige Benachrichtigung erfolgt ist, seit Beginn der Protokollierung erfassten Einträge an. Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt.

7.85 Reliability

Datentyp	BACnetReliability
Datentyp IEC	BACnetReliability
In Objekt	Accumulator, Analog-Value, Binary-Value, Life-Safety-Point, Life-Safety-Zone, Multi-State-Value, Pulse-Converter, Schedule
Beschreibung	In Objekt Accumulator: Die Eigenschaft Reliability des Typs BACnetReliability zeigt an, ob der Wert für den Present_Value oder die Steuerung des physikalischen Einganges „zuverlässig“ ist, soweit das BACnet-Gerät oder der Anwender dies feststellen können oder warum dies nicht der Fall ist. Die Eigenschaft Reliability für dieses Objekt kann folgende Werte annehmen: {NO_FAULT_DETECTED, NO_SENSOR, OVER_RANGE, UNDER_RANGE, OPEN_LOOP, SHORTED_LOOP, UNRELIABLE_OTHER}

In Objekt Analog-Value:

Die Eigenschaft Reliability des Typs BACnetReliability zeigt an, ob der Wert für den Present_Value „zuverlässig“ ist, soweit das BACnet-Gerät dies feststellen können oder warum dies nicht der Fall ist. Die Eigenschaft Reliability für dieses Objekt kann folgende Werte annehmen:

{NO_FAULT_DETECTED, OVER_RANGE,
UNDER_RANGE, UNRELIABLE_OTHER}

In Objekt Binary-Value:

Die Eigenschaft Reliability des Typs BACnetReliability zeigt an, ob der Wert für den Present_Value „zuverlässig“ ist, soweit das BACnet-Gerät oder der Anwender dies feststellen können oder warum dies nicht der Fall ist. Die Eigenschaft Reliability für dieses Objekt kann folgende Werte annehmen:

{NO_FAULT_DETECTED, UNRELIABLE_OTHER}

In Objekt Life-Safety-Point:

Die Eigenschaft Reliability des Typs BACnetReliability zeigt an, ob der Wert für Present_Value oder die Steuerung physikalischer Eingänge „zuverlässig“ ist, soweit das BACnet-Gerät oder der Anwender dies feststellen können oder warum dies nicht der Fall ist. Die Eigenschaft Reliability für dieses Objekt kann folgende Werte annehmen:

{NO_FAULT_DETECTED, OPEN_LOOP, SHORTED_LOOP,
MULTI_STATE_FAULT, UNRELIABLE_OTHER}

In Objekt Life-Safety-Zone:

Die Eigenschaft Reliability des Typs BACnetReliability zeigt an, ob der Wert für Present_Value oder die Steuerung physikalischer Eingänge „zuverlässig“ ist, soweit das BACnet-Gerät oder der Anwender dies feststellen können oder warum dies nicht der Fall ist. Die Eigenschaft Reliability für dieses Objekt kann folgende Werte annehmen:

{NO_FAULT_DETECTED, OPEN_LOOP, SHORTED_LOOP,
MULTI_STATE_FAULT, UNRELIABLE_OTHER}

Bedingungen für die Generierung eines TO-FAULT-Ereignisses:

- das Flag TO-FAULT muss in der Eigenschaft Event_Enable aktiviert sein und
- der Present_Value muss mindestens einem der in der Liste Fault_Values enthaltenen Werte entsprechen.

In Objekt Multi-State-Value:

Die Eigenschaft Reliability des Typs BACnetReliability zeigt an, ob der Wert für Present_Value „zuverlässig“ ist, soweit das BACnet-Gerät oder der Anwender dies feststellen können oder warum dies nicht der Fall ist. Die Eigenschaft Reliability ist erforderlich, wenn die Eigenschaft Fault_Values vorhanden ist. Die Eigenschaft Reliability für dieses Objekt kann folgende Werte annehmen:

{NO_FAULT_DETECTED, MULTI_STATE_FAULT,
UNRELIABLE_OTHER}

In Objekt Pulse-Converter:

Die Eigenschaft Reliability des Typs BACnetReliability zeigt an, ob der Present_Value und/oder Count-Eigenschaften oder die Steuerung des physikalischen Einganges „zuverlässig“ sind, soweit das BACnet-Gerät oder der Anwender dies feststellen können oder warum dies nicht der Fall ist. Die Eigenschaft Reliability für dieses Objekt kann folgende Werte annehmen:

{NO_FAULT_DETECTED, NO_SENSOR, OVER_RANGE,
UNDER_RANGE, OPEN_LOOP, SHORTED_LOOP,
UNRELIABLE_OTHER, CONFIGURATION_ERROR}

Verweist Input_Reference auf eine Eigenschaft, die nicht den Datentyp Unsigned oder Integer hat bzw. nicht als Eingangsquelle für dieses Objekt unterstützt wird, gibt die Eigenschaft Reliability CONFIGURATION_ERROR an.

In Objekt Schedule:

Die Eigenschaft Reliability des Typs BACnetReliability bietet zeigt an, ob die Eigenschaften des Schedule-Objektes konstant sind. Alle Werte in den Eigenschaften Weekly_Schedule, Exception_Schedule und Schedule_Default, die nicht „NULL“ sind, müssen denselben Datentyp haben, und alle Mitglieder der List_Of_Object_Property_References müssen mit diesem Datentyp schreibbar sein. Werden diese Bedingungen nicht erfüllt, hat diese Eigenschaft den Wert CONFIGURATION_ERROR.

Die Eigenschaft Reliability für dieses Objekt kann folgende Werte annehmen:

{NO_FAULT_DETECTED, CONFIGURATION_ERROR,
UNRELIABLE_OTHER}

Falls List_Of_Object_Property_References ein Mitglied enthält, das auf eine Eigenschaft in einem entfernten Gerät verweist, kann sich die Erkennung eines Konfigurationsfehlers bis zum Versuch, einen Schedule-Wert zu schreiben, verzögern.

7.86 Relinquish_Default

Datentyp REAL

Datentyp IEC REAL

In Objekt Analog-Value

Beschreibung Diese Eigenschaft ist der Standardwert für die Eigenschaft Present_Value, wenn alle Prioritätswerte für Befehle der Eigenschaft Priority_Array den Wert „NULL“ haben. Wenn die Eigenschaft Relinquish_Default oder Priority_Array vorhanden ist, müssen beide vorhanden sein. Kann Present_Value Befehle empfangen, müssen die Eigenschaften Priority_Array und Relinquish_Default vorhanden sein.

Datentyp BACnetBinaryPV

Datentyp IEC BACnetBinaryPV

In Objekt Binary-Value

Beschreibung Diese Eigenschaft ist der Standardwert für die Eigenschaft Present_Value, wenn alle Prioritätswerte für Befehle der Eigenschaft Priority_Array den Wert „NULL“ haben. Wenn die Eigenschaft Relinquish_Default oder Priority_Array vorhanden ist, müssen beide vorhanden sein. Kann Present_Value Befehle empfangen, müssen die Eigenschaften Priority_Array und Relinquish_Default vorhanden sein und Present_Value muss schreibbar sein.

Datentyp	Unsigned
Datentyp IEC	WORD
In Objekt	Multi-State-Value
Beschreibung	Diese Eigenschaft ist der Standardwert für die Eigenschaft Present_Value, wenn alle Prioritätswerte für Befehle der Eigenschaft Priority_Array den Wert „NULL“ haben. Wenn die Eigenschaft Relinquish_Default oder Priority_Array vorhanden ist, müssen beide vorhanden sein. Kann Present_Value Befehle empfangen, müssen die Eigenschaften Priority_Array und Relinquish_Default vorhanden sein.

7.87 Scale

Datentyp	BACnetScale
Datentyp IEC	BACnetScale
In Objekt	Accumulator
Beschreibung	Diese Eigenschaft des Typs BACnetScale gibt den Konvertierungsfaktor zur Multiplikation mit dem Wert der Eigenschaft Present_Value an, um einen Wert mit den in der Eigenschaft Units angegebenen Einheiten anzuzeigen. Die Optionen dieser Eigenschaft bestimmen die Skalierung, die der Client ausführt, der dieses Objekt ausliest:

Tabelle 71: Optionen des Typs BACnetScale

Option	Datentyp	Angezeigter Wert in Units
floatScale	REAL	Present_Value × Scale
integerScale	INTEGER	Present_Value × 10 ^{Scale}

7.88 Scale_Factor

Datentyp	REAL
Datentyp IEC	REAL
In Objekt	Pulse-Converter
Beschreibung	Diese Eigenschaft des Typs REAL gibt den Konvertierungsfaktor zur Berechnung des Present_Value an. Sie stellt die Änderung des Present_Value bei einer Änderung um „1“ von Count dar.

7.89 Schedule_Default

Datentyp	beliebig
Datentyp IEC	ARRAY [0...10] of BYTE
In Objekt	Schedule
Beschreibung	Diese Eigenschaft enthält einen Standardwert für die Eigenschaft Present_Value, wenn kein anderer Schedule-Wert gültig ist (siehe Kapitel „Present_Value“). Der Wert kann ein einfacher Datentyp sein. Wird in die Eigenschaft Schedule_Default ein Wert mit einem von dieser Instanz des Schedule-Objektes nicht unterstützten Datentyp geschrieben (z. B. kann die Eigenschaft List_Of_Object_Property_References nicht für einen Verweis auf eine Eigenschaft mit einem nicht unterstützten Datentyp konfiguriert werden), gibt das Gerät unter Umständen Result(-) mit der Fehlerklasse PROPERTY und dem Fehlercode DATATYPE_NOT_SUPPORTED zurück.

7.90 Setting

Datentyp	Unsigned8
Datentyp IEC	BYTE
In Objekt	Life-Safety-Point
Beschreibung	Diese optionale Eigenschaft des Typs Unsigned8 gibt die gewünschte Einstellung für die Eingänge oder den Prozess an, die zur Bestimmung des logischen Zustandes der Eigenschaft Present_Value verwendet werden. Der Wertebereich für die Eigenschaft Setting reicht von „0“ (wenig empfindlich) bis „100“ (sehr empfindlich). Die Bewertung der Einstellungen und der geeignete Wert für ein bestimmtes Life-Safety-Point-Objekt sind lokal unterschiedlich.

7.91 Silenced

Datentyp	BACnetSilencedState
Datentyp IEC	BACnetSilencedState
In Objekt	Life-Safety-Point, Life-Safety-Zone
Beschreibung	Diese Eigenschaft des Typs BACnetSilencedState gibt an, ob beim letzten Wechsel dieses Objektes, der zu einer akustischen oder optischen Anzeige geführt hat, diese durch den Empfang einer LifeSafetyOperation-Dienstanfrage oder durch einen lokalen Prozess aufgehoben wurde.

7.92 Start_Time

Datentyp	BACnetDateTime
Datentyp IEC	BACnetDateTime
In Objekt	Trend-Log
Beschreibung	Diese Eigenschaft des Typs BACnetDateTime gibt Zeit und Datum an, ab dem die Protokollierung für diese Eigenschaft aktiviert ist. Enthalten Felder für BACnetDateTime Platzhalter, werden die Angaben zur Protokollierung in Start_Time ignoriert. Gibt Start_Time ein Datum und eine Zeit nach Stop_Time an, wird die Protokollierung deaktiviert. Falls diese Eigenschaft vorhanden ist, muss sie schreibbar sein.

7.93 State_Text

Datentyp	BACnetARRAY [N] of CharacterString
Datentyp IEC	ARRAY [1...3] of STRING;
In Objekt	Multi-State-Value
Beschreibung	Diese Eigenschaft ist ein BACnetARRAY aus Zeichenketten, die alle möglichen Zustände der Eigenschaft Present_Value darstellen. Die Anzahl der Beschreibungen entspricht der in der Eigenschaft Number_Of_States angegebenen Anzahl verschiedener Zustände. Der Present_Value wird als Integer interpretiert und dient als Index für das Feld. Bei Änderung der Größe dieses Feldes wird die Eigenschaft Number_Of_States auf denselben Wert geändert.

7.94 Status_Flags

Datentyp	BACnetStatusFlags
Datentyp IEC	BACnetStatusFlags
In Objekt	Accumulator, Analog-Value, Binary-Value, Life-Safety-Point, Life-Safety-Zone, Multi-State-Value, Pulse-Converter, Schedule

Beschreibung Diese Eigenschaft des Typs BACnetStatusFlags stellt vier Boolean-Flags dar, welche den allgemeinen Zustand obiger Objekte darstellen. Drei der Flags sind mit den Werten anderer Eigenschaften dieses Objektes verknüpft. Ein Zustand mit weiteren Details kann durch Auslesen der mit diesen Flags verknüpften Eigenschaften ermittelt werden. Die Beziehung zwischen einzelnen Flags wird nicht durch das Protokoll definiert. Die vier Flags sind:

{IN_ALARM, FAULT, OVERRIDDEN, OUT_OF_SERVICE}

wobei:

IN_ALARM

logisch „FALSE“ („0“) ist, wenn die Eigenschaft Event_State den Wert NORMAL hat, andernfalls logisch „TRUE“ (1). Für das Schedule-Objekt ist der Wert für dieses Flag logisch „FALSE“ („0“).

FAULT

logisch „TRUE“ (1) ist, wenn die Eigenschaft Reliability vorhanden ist und nicht den Wert NO_FAULT_DETECTED hat, andernfalls logisch „FALSE“ („0“).

OVERRIDDEN**In Objekt Accumulator und Multi-State Input:**

logisch „TRUE“ („1“) ist, wenn der Punkt von einem lokalen Mechanismus des BACnet-Gerätes außer Kraft gesetzt wurde. In diesem Fall bedeutet „Overridden“, dass die Eigenschaften Present_Value und Reliability Änderungen am physikalischen Eingang nicht mehr verfolgen. Andernfalls ist der Wert logisch „FALSE“ („0“).

in Objekt Analog-Value, Binary-Value und Schedule:

logisch „TRUE“ („1“) ist, wenn das Objekt von einem lokalen Mechanismus des BACnet-Gerätes außer Kraft gesetzt wurde. In diesem Fall bedeutet „Overridden“, dass die Eigenschaft Present_Value nicht durch BACnet-Dienste verändert werden kann. Andernfalls ist der Wert logisch „FALSE“ („0“).

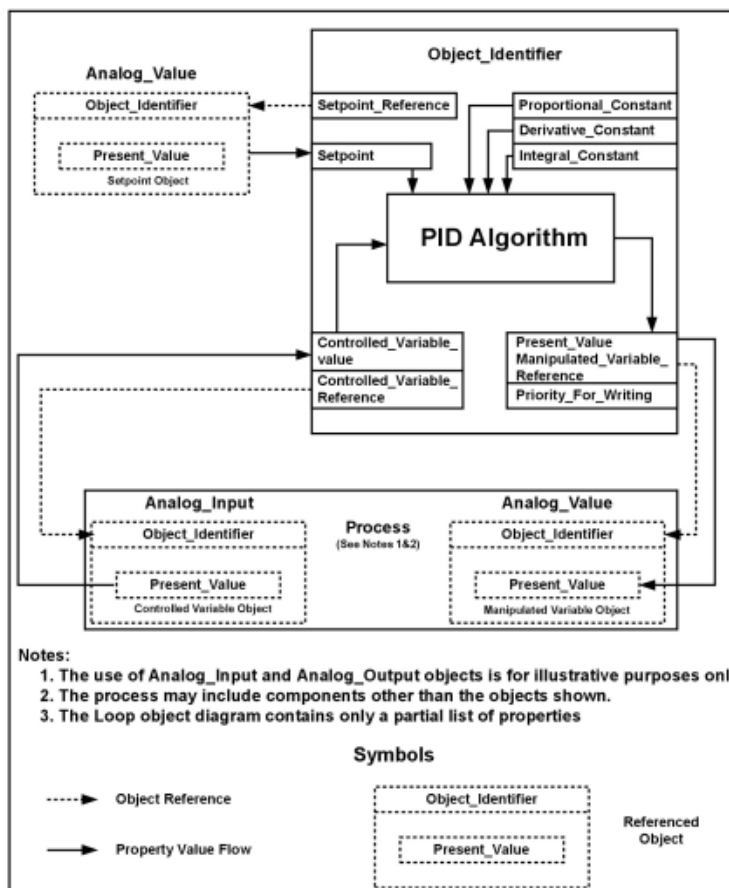


Abbildung 7: Loop-Objektstruktur mit referenzierten Objekten

In Objekt Life-Safety-Point und Life-Safety-Zone:

logisch „TRUE“ („1“) ist, die Anwendung von einem lokalen Mechanismus des BACnet-Gerätes außer Kraft gesetzt wurde. In diesem Fall bedeutet „Overridden“, dass die Eigenschaften Present_Value und Reliability den physikalischen Eingang nicht mehr darstellen. Andernfalls ist der Wert logisch „FALSE“ („0“).

In Objekt Multi-State-Value:

logisch „TRUE“ („1“) ist, wenn die Anwendung von einem lokalen Mechanismus des BACnet-Gerätes außer Kraft gesetzt wurde. In diesem Fall bedeutet „Overridden“, dass die Eigenschaften Present_Value und Reliability den physikalischen Ausgang nicht mehr darstellen. Andernfalls ist der Wert logisch „FALSE“ („0“).

In Objekt Pulse-Converter:

logisch „TRUE“ (1) ist, wenn die Anwendung von einem lokalen Mechanismus des BACnet-Gerätes außer Kraft gesetzt wurde.

OUT_OF_SERVICE

Logisch „TRUE“ (1) wenn die Eigenschaft Out_Of_Service „TRUE“ ist, sonst logisch „FALSE“ („0“).

7.95 Stop_Time

Datentyp BACnetDateTime

Datentyp IEC BACnetDateTime

In Objekt Trend-Log

Beschreibung Diese Eigenschaft des Typs BACnetDateTime gibt Zeit und Datum an, ab dem die Protokollierung für diese Eigenschaft deaktiviert ist. Enthalten Felder für BACnetDateTime Platzhalter, werden die Angaben zur Protokollierung in Stop_Time ignoriert. Gibt Stop_Time ein Datum und eine Zeit vor Start_Time an, wird die Protokollierung deaktiviert. Falls diese Eigenschaft vorhanden ist, muss sie schreibbar sein.

7.96 Stop_When_Full

Datentyp BOOLEAN

Datentyp IEC BOOL

In Objekt Trend-Log

Beschreibung Diese Eigenschaft des Typs BOOLEAN gibt an, ob die Protokollierung bei vollem Speicher abgebrochen wird („TRUE“) oder nicht („FALSE“). Bei Beenden der Protokollierung wird Log_Enable auf „FALSE“ gesetzt.

7.97 Time_Delay

Datentyp Unsigned

Datentyp IEC WORD

In Objekt Accumulator, Analog-Value, Binary-Value, Life-Safety-Point, Life-Safety-Zone, Multi-State-Value, Pulse-Converter

Beschreibung **In Objekt Accumulator:**
Diese Eigenschaft des Typs Unsigned gibt den Mindestzeitraum in Sekunden an, den Pulse_Rate außerhalb des durch die Eigenschaften High_Limit und Low_Limit definierten Bereiches sein muss, bevor ein TO-OFFNORMAL- oder TO-NORMAL-Ereignis generiert wird. Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt.

In Objekt Analog-Value:

Diese Eigenschaft des Typs Unsigned gibt den Mindestzeitraum in Sekunden an, den der Present_Value außerhalb des durch die Eigenschaften High_Limit und Low_Limit definierten Bereiches sein muss, bevor ein TO-OFFNORMAL-Ereignis generiert wird bzw. innerhalb des Bereiches einschließlich der Eigenschaft Deadband, bevor ein TO-NORMAL-Ereignis generiert wird.

In Objekt Binary-Value:

Diese Eigenschaft des Typs Unsigned gibt den Mindestzeitraum in Sekunden an, den der Present_Value nicht gleich der Eigenschaft Alarm_Value sein muss, bevor ein TO-OFFNORMAL-Ereignis generiert wird bzw. gleich der Eigenschaft Alarm_Value, bevor ein TO-NORMAL-Ereignis generiert wird. Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt.

In Objekt Life-Safety-Point und Life-Safety-Zone:

Diese optionale Eigenschaft des Typs Unsigned gibt den Mindestzeitraum in Sekunden an, den der Present_Value folgende Werte annehmen muss:

- gleich einem der Werte der Eigenschaft Life_Safety_Alarm_Values, bevor ein TO-OFFNORMAL-Ereignis generiert wird, oder
- gleich einem der Werte der Eigenschaft Alarm_Values, bevor ein TO-OFFNORMAL-Ereignis generiert wird, oder
- nicht gleich einem der Werte der Eigenschaft Life_Safety_Alarm_Values, Alarm_Values oder Fault_Values, bevor ein TO-NORMAL-Ereignis generiert wird.

Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt.

In Objekt Multi-State-Value:

Diese Eigenschaft des Typs Unsigned gibt den Mindestzeitraum in Sekunden an, den der Present_Value folgende Werte annehmen muss:

- gleich einem der Werte der Eigenschaft Alarm_Values, bevor ein TO-OFFNORMAL-Ereignis generiert wird, oder
- nicht gleich einem der Werte der Eigenschaft Alarm_Values, bevor ein TO-NORMAL-Ereignis generiert wird.

Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt.

In Objekt Pulse-Converter:

Diese Eigenschaft des Typs Unsigned gibt den Mindestzeitraum in Sekunden an, den der Present_Value außerhalb des durch die Eigenschaften High_Limit und Low_Limit definierten Bereiches sein muss, bevor ein TO-OFFNORMAL-Ereignis generiert wird bzw. innerhalb des Bereiches einschließlich der Eigenschaft Deadband, bevor ein TO-NORMAL-Ereignis generiert wird. Diese Eigenschaft ist erforderlich, wenn das Objekt objektinternes Melden unterstützt.

7.98 Time_Of_Active_Time_Reset

Datentyp	BACnetDateTime
Datentyp IEC	BACnetDateTime
In Objekt	Binary-Value
Beschreibung	Diese Eigenschaft des Typs BACnetDateTime gibt Datum und Zeit des letzten Zurücksetzens der Eigenschaft Elapsed_Active_Time auf Null an. Wenn eine der optionalen Eigenschaften Elapsed_Active_Time oder Time_Of_Active_Time_Reset vorhanden ist, müssen beide verwendet werden.

7.99 Time_Of_State_Count_Reset

Datentyp	BACnetDateTime
Datentyp IEC	BACnetDateTime
In Objekt	Binary-Value
Beschreibung	Diese Eigenschaft des Typs BACnetDateTime gibt Zeit und Datum des letzten Zurücksetzens der Eigenschaft Change_Of_State_Count auf Null an. Falls eine der optionalen Eigenschaften Change_Of_State_Time, Change_Of_State_Count oder Time_Of_State_Count_Reset vorhanden ist, müssen alle diese Eigenschaften vorhanden sein.

7.100 Total_Record_Count

Datentyp	Unsigned32
Datentyp IEC	DWORD
In Objekt	Trend-Log
Beschreibung	Diese Eigenschaft des Typs Unsigned32 gibt die Gesamtanzahl der seit Erstellung des Trend-Log-Objektes gesammelten Records an. Erreicht der Wert des Total_Record_Count den höchsten möglichen Wert von „ $2^{32} - 1$ “, ist der nächste Wert „1“. Ist der Zähler auf „1“ zurückgesprungen, kann die Gesamtanzahl der Records nicht mehr ermittelt werden, die Funktion zur Generierung von Benachrichtigungen bleibt jedoch erhalten.

7.101 Tracking_Value

Datentyp	BACnetLifeSafetyState
Datentyp IEC	BACnetLifeSafetyState
In Objekt	Life-Safety-Point, Life-Safety-Zone
Beschreibung	Diese optionale Eigenschaft des Typs BACnetLifeSafetyState gibt den nicht-verknüpften Zustand des Life-Safety-Point-/Life-Safety-Zone-Objektes an. Die Methode für die Ermittlung des Zustandes ist lokal unterschiedlich. Im Gegensatz zur Eigenschaft Present_Value, die bis zum Zurücksetzen NON-NORMAL-Zustände annehmen kann, verfolgt Tracking_Value Zustandsänderungen kontinuierlich.

7.102 Units

Datentyp	BACnetEngineeringUnits
Datentyp IEC	BACnetEngineeringUnits
In Objekt	Accumulator, Analog-Value, Life-Safety-Point, Pulse-Converter
Beschreibung	In Objekt Accumulator: Diese Eigenschaft des Typs BACnetEngineeringUnits gibt die Maßeinheiten für Present_Value bei Multiplikation mit dem durch Scale angegebenen Skalierungsfaktor an. In Objekt Analog-Value: Diese Eigenschaft des Typs BACnetEngineeringUnits gibt die Maßeinheiten für dieses Objekt an. In Objekt Life-Safety-Point: Diese Eigenschaft des Typs BACnetEngineeringUnits gibt die durch die Eigenschaft Direct_Reading dargestellten Maßeinheiten an. Ist diese Eigenschaft vorhanden, muss auch die Eigenschaft Direct_Reading verwendet werden. In Objekt Pulse-Converter: Diese Eigenschaft des Typs BACnetEngineeringUnits gibt die Maßeinheiten für die Eigenschaft Present_Value an.

Information**Weitere Information**

Die Liste der standardisierten Engineering-Einheiten finden Sie in der Dokumentation der BACnet-Bibliothek „BACnet_xx.lib“ auf der Internetseite <http://www.wago.com> unter Service → Downloads → Gebäudeautomation → BACnet Downloads → Software

7.103 Update_Time

Datentyp BACnetDateTime

Datentyp IEC BACnetDateTime

In Objekt Pulse-Converter

Beschreibung Diese nur lesbare Eigenschaft des Typs BACnetDateTime stellt Datum und Zeit der letzten Änderung der Eigenschaft Count als Ergebnis einer Akkumulierung von Eingangsimpulsen dar und wird gemeinsam mit der Eigenschaft Count aktualisiert. Hat noch keine Änderung stattgefunden, hat die Eigenschaft Platzhalter für alle Datums- und Zeitfelder.

7.104 Valid_Samples

Datentyp Unsigned

Datentyp IEC WORD

In Objekt Averaging

Beschreibung Diese nur lesbare Eigenschaft des Typs Unsigned stellt die Anzahl der Samples dar, die für das aktuelle Fenster erfolgreich abgefragt wurden. Mit diesem Wert kann bestimmt werden, ob im aktuellen Window_Interval Samples fehlen. Die Anzahl fehlender Samples im aktuellen Fenster kann durch Abziehen der Valid_Samples von Attempted_Samples ermittelt werden. Ein Wert größer „Null“ gibt die Anzahl der Samples an, bei denen bei der Abfrage ein Fehler aufgetreten ist. Nach einem Neustart des Gerätes oder nach dem Schreiben in Attempted_Samples, Object_Property_Reference, Window_Samples oder Window_Interval mit BACnet-Diensten vor einer Abfrage hat Valid_Samples den Wert „Null“.

7.105 Value_Before_Change

Datentyp Unsigned

Datentyp IEC WORD

In Objekt Accumulator

Beschreibung Diese Eigenschaft des Typs Unsigned gibt den Wert der Eigenschaft Present_Value vor dem letzten Schreiben in die Eigenschaft Value_Set oder Value_Before_Change an. Ist noch kein Schreibvorgang erfolgt, hat diese Eigenschaft den Wert „Null“. Falls diese Eigenschaft vorhanden ist, muss die Eigenschaft Value_Set nur lesbar sein. Ist diese Eigenschaft schreibbar, werden die nachfolgenden Aktionen, für die entsprechende Eigenschaften vorhanden sind, bei Schreiben der Eigenschaft durch das Objekt gemeinsam durchgeführt:

- Der Wert von Present Value wird in die Eigenschaft Value_Set kopiert.
- Der in Value_Before_Change geschriebene Wert wird in der Eigenschaft Value_Before_Change gespeichert.
- Die aktuelle Zeit und das Datum werden in der Eigenschaft Value_Change_Time gespeichert.

Hinweis



Änderung durch andere Aktionen vermeiden!

Bei Ausführung dieser Aktionen ist es äußerst wichtig, dass kein anderer Prozess die Eigenschaften Present_Value, Value_Set und Value_Before_Change ändert.

7.106 Value_Change_Time

Datentyp BACnetDateTime

Datentyp IEC BACnetDateTime

In Objekt Accumulator

Beschreibung Diese nur lesbare Eigenschaft des Typs BACnetDateTime muss vorhanden sein, wenn mit den Eigenschaften Value_Before_Change oder Value_Set in die Eigenschaft Present_Value geschrieben werden kann. Die Eigenschaft gibt Datum und Zeit der letzten derartigen Schreibaktion an. Ist noch kein Schreibvorgang erfolgt, hat diese Eigenschaft Platzhalter für alle Datums- und Zeitfelder.

7.107 Value_Set

Datentyp Unsigned

Datentyp IEC WORD

In Objekt Accumulator

Beschreibung Diese Eigenschaft des Typs Unsigned gibt den Wert der Eigenschaft Present_Value nach dem letzten Schreiben in die Eigenschaften Value_Set oder Value_Before_Change an. Ist noch kein Schreibvorgang erfolgt, hat diese Eigenschaft den Wert „Null“. Ist diese Eigenschaft schreibbar, muss die Eigenschaft Value_Before_Change nur lesbar sein. Ist diese Eigenschaft schreibbar, werden die nachfolgenden Aktionen, für die entsprechende Eigenschaften vorhanden sind, bei Schreiben der Eigenschaft durch das Objekt gemeinsam durchgeführt:

- Der Wert von Present_Value wird in die Eigenschaft Value_Before_Change kopiert.
- Der in Value_Set geschriebene Wert wird in den Eigenschaften Value_Set und Present_Value gespeichert.
- Die aktuelle Zeit und das Datum werden in der Eigenschaft Value_Change_Time gespeichert.

Hinweis



Änderung durch andere Aktionen vermeiden!

Bei Ausführung dieser Aktionen ist es äußerst wichtig, dass kein anderer Prozess die Eigenschaften Present_Value, Value_Set und Value_Before_Change ändert.

7.108 Weekly_Schedule

Datentyp BACnetARRAY [7] of BACnetDailySchedule

Datentyp IEC ARRAY [1...7] of BACnetDailySchedule

In Objekt Schedule

Beschreibung Diese Eigenschaft ist ein BACnetARRAY mit genau sieben Elementen. Jedes der Elemente 1 bis 7 enthält einen BACnetDailySchedule. Ein BACnetDailySchedule besteht aus einer „List of BACnetTimeValues“, also Zeit-/Wertpaaren, die den gesamten Ablauf der Aktionen des Zeitplans an einem Wochentag beschreiben, an dem kein Exception_Schedule gilt. Die Feldelemente 1 bis 7 entsprechen den Tagen Montag bis Sonntag. Weekly_Schedule ist eine optionale Eigenschaft, aber entweder Weekly_Schedule oder Exception_Schedule mit gültigen Einträgen muss für jede Instanz eines Schedule-Objektes verwendet werden.

Wird die Eigenschaft Weekly_Schedule mit einem Schedule-Element geschrieben, das einen von dieser Instanz des Schedule-Objektes nicht unterstützten Datentyp enthält (z. B. kann die Eigenschaft List_Of_Object_Property_References nicht so konfiguriert werden, dass sie auf eine Eigenschaft des nicht unterstützten Datentyps verweist), gibt das Gerät unter Umständen Result(-) mit der Fehlerklasse PROPERTY und dem Fehlercode DATATYPE_NOT_SUPPORTED zurück.

7.109 Window_Interval

Datentyp Unsigned

Datentyp IEC WORD

In Objekt Averaging

Beschreibung Diese Eigenschaft des Typs Unsigned gibt den Zeitraum in Sekunden für die Berechnung der Minimal-, Maximal und Durchschnittswerte an. Der niedrigste zulässige Wert für Window_Interval ist lokal unterschiedlich. Alle Window_Interval geteilt durch Window_Samples Sekunden wird ein neues Sample durch Auslesen des Wertes der durch Object_Property_Reference referenzierten Eigenschaft abgefragt. Ob das Sample eine Momentaufnahme oder einen durchgehenden, berechneten Wert darstellt, ist lokal unterschiedlich. Wird mit BACnet-Diensten in Window_Interval geschrieben, werden alle Samples im Speicher ungültig, Attempted_Samples und Valid_Samples erhalten den Wert „Null“, Minimum_Value den Wert „INF“, Average_Value NaN und Maximum_Value „-INF“.

7.110 Window_Samples

Datentyp	Unsigned
Datentyp IEC	WORD
In Objekt	Averaging
Beschreibung	Diese Eigenschaft des Typs Unsigned gibt die Anzahl der Samples an, die in dem Zeitraum erfasst werden, den die Eigenschaft Window_Interval angibt. Window_Samples muss größer „Null“ sein, alle Implementierungen müssen mindestens 15 Samples unterstützen. Alle Window_Interval geteilt durch Window_Samples Sekunden wird ein neues Sample durch Auslesen des Wertes der durch Object_Property_Reference referenzierten Eigenschaft abgefragt. Ob das Sample eine Momentaufnahme oder einen durchgehenden, berechneten Wert darstellt, ist lokal unterschiedlich. Wird mit BACnet-Diensten in Window_Samples geschrieben, werden alle Samples im Speicher ungültig, Attempted_Samples und Valid_Samples erhalten den Wert „Null“, Minimum_Value den Wert „INF“, Average_Value „NaN“ und Maximum_Value „-INF“.

7.111 Zone_Members

Datentyp	List of BACnetDeviceObjectReference
Datentyp IEC	ARRAY [0...3] of BACnetDeviceObjectReference
In Objekt	Life-Safety-Zone
Beschreibung	Diese Eigenschaft des Typs „List of BACnetDeviceObjectReference“ gibt die Life-Safety-Point- und Life-Safety-Zone-Objekte an, die Mitglieder der durch dieses Objekt dargestellten Zone sind. Diese Eigenschaft kann auf die Unterstützung von Verweisen auf interne Objekte des Gerätes mit dem Life-Safety-Zone-Objekt beschränkt werden. Ist die Eigenschaft schreibbar und auf Verweise zu Objekten beschränkt, die im Gerät enthalten sind, führt der Versuch, einen Verweis auf ein Objekt außerhalb des Gerätes in diese Eigenschaft zu schreiben, zur Ausgabe von Result(-) mit der Fehlerklasse PROPERTY und dem Fehlercode OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED.

8 Gesamtliste der BACnet-Dienst-Parameter

Die Parameter der BACnet-Dienste werden nachfolgend in alphabetischer Reihenfolge aufgeführt. Für jeden Parameter werden die Dienste aufgelistet, in welchen der Parameter vorhanden ist. Die Beschreibung gibt nähere Auskunft über die Funktion des Parameters.

Eventuell sind Parameter oder Parametergruppen in anderen Parametern enthalten. Darauf weist der Satz „referenziert durch Dienst ... in Parameter ...“ hin. In der Beschreibung selbst wird der Name des beinhaltenden Parameters und bei Bedarf die Parametergruppe genannt.

Dienste können einen Parameter mehrfach besitzen, wenn dieser in unterschiedlichen Parametergruppen vorkommen.

8.1 Acknowledged Transitions

Dienste GetAlarmSummary, GetEventInformation

Beschreibung Dieser Parameter des Typs BACnetEventTransitionBits gibt den Wert der Eigenschaft Acked_Transitions des Objektes an.

8.2 Acknowledging Process Identifier

Dienste AcknowledgeAlarm

Beschreibung Dieser Parameter des Typs Unsigned32 gibt den Parameter „Process Identifier“ der Ereignisbenachrichtigung an, den diese Bestätigung beantwortet. So kann das initiiierende Objekt sicherstellen, dass der entsprechende Prozess die Benachrichtigung erhalten hat.

8.3 Acknowledgment Filter

Dienste GetEnrollmentSummary

Beschreibung Dieser Parameter des Typs Unsigned32 gibt den Parameter „Process Identifier“ der Ereignisbenachrichtigung an, den diese Bestätigung beantwortet. So kann das initiiierende Objekt sicherstellen, dass der entsprechende Prozess die Benachrichtigung erhalten hat.

ALL Fordert an, dass die zurückgegebene Zusammenfassung alle Objekte enthält, die Ereignisse auslösen, ohne Rücksicht darauf, ob ein Objekt über eine Bestätigung verfügt oder nicht.

ACKED Fordert an, dass die zurückgegebene Zusammenfassung nur die Objekte enthält, deren Eigenschaft Acked_Transitions an jeder Bitposition den Wert „1“ hat.

NOT-ACKED Fordert an, dass die zurückgegebene Zusammenfassung nur die Objekte enthält, deren Eigenschaft Acked_Transitions an einer oder mehr Bitpositionen den Wert „0“ hat.

8.4 Acknowledgment Source

Dienste AcknowledgeAlarm

Beschreibung Dieser Parameter des Typs CharacterString gibt die Identität des Anwenders oder Prozesses an, der die Ereignisbenachrichtigung bestätigt.

8.5 AckRequired

Dienste ConfirmedEventNotification, UnconfirmedEventNotification

Beschreibung Dieser Parameter des Typs BOOLEAN gibt an, ob diese Benachrichtigung eine Bestätigung erfordert („TRUE“) oder nicht („FALSE“). Dieser Parameter darf nur vorhanden sein, wenn der Parameter „Notify Type“ EVENT oder ALARM ist.

8.6 Alarm State

Dienste GetAlarmSummary

Beschreibung Dieser Parameter des Typs BACnetEventState gibt den Wert der Eigenschaft Event_State des Objektes an.

8.7 Argument

Dienste AcknowledgeAlarm, AtomicReadFile, AtomicWriteFile, ConfirmedCOVNotification, ConfirmedEventNotification, DeviceCommunicationControl, GetAlarmSummary, GetEnrollmentSummary, GetEventInformation, I-Am, I-Have, LifeSafetyOperation, Object Access, ReadProperty, ReadPropertyConditional, ReadPropertyMultiple, ReadRange, ReinitializeDevice, RemoveListElement, SubscribeCOV, SubscribeCOVProperty, TimeSynchronization, UnconfirmedCOVNotification, UnconfirmedEventNotification, UnconfirmedPrivateTransfer, UnconfirmedTextMessage, UTCTimeSynchronization, Who-Has, Who-Is, WriteProperty, WritePropertyMultiple

Beschreibung Dieser Parameter gibt die Parameter für obige Dienstanfragen an.

8.8 By Position

Dienste Referenziert durch ReadRange im Parameter Range

Beschreibung Der Parameter „By Position“ gibt an, dass ein Index auf die auszulesenden Elemente verweist.

8.9 By Sequence Number

Dienste Referenziert durch ReadRange im Parameter Range

Beschreibung Der Parameter „By Sequence Number“ gibt an, dass auf die auszulesenden Elemente mit einer Sequenznummer verwiesen wird und dass die Antwort die Sequenznummer des ersten zurückgegebenen Elementes enthalten muss. Dies ist ein semantischer Unterschied zum Parameter „By Position“. Die in „By Position“ angegebene „Reference Number“ verweist auf ein Element an einer bestimmten Position der Liste. Im Gegensatz dazu verweist die „Reference Number“ von „By Sequence Number“ auf ein Element mit einer bestimmten Sequenznummer, die bei Hinzufügen des Elementes zur Liste vergeben wird. Nicht alle Listen verfügen über Sequenznummern. Die Eigenschaft Log_Buffer des Trend-Log-Objektes ist ein Beispiel für eine Liste mit Sequenznummern.

8.10 By Time

Dienste Referenziert durch ReadRange im Parameter Range

Beschreibung Der Parameter „By Time“ gibt an, dass ein Zeitstempel auf das auszulesende Element verweist und dass die Sequenznummer des Elementes in der Antwort enthalten sein muss. Dieses Format wird erwartet, wenn Listen durchsucht werden, die nach der Zeit indiziert sind.

8.11 Comparison Value

Dienste	Referenziert durch ReadPropertyConditional im Parameter „Object Selection Criteria“
Beschreibung	Dieser Parameter ist eine Konstante des selben Datentyps wie die zu vergleichende Eigenschaft. Bei Verwendung von Comparison Value zur Auswahl von Eigenschaften des Typs CharacterString können die Platzhalter „?“ und „*“ nur für die Operatoren „=“ und „≠“ verwendet werden. Das Zeichen „?“ ist ein Platzhalter für exakt ein Zeichen an der Stelle, an der es in der Zeichenkette enthalten ist. In einem „Comparison Value“ kann „?“ mehrmals auftreten. Folglich entspricht „??1“ „AC1“, „CP1“ und „SF1“ überein, jedoch nicht „C1“, „P1“ oder „PUMP1“. Das Zeichen „*“ ist ein Platzhalter für Null oder mehr Zeichen ab der entsprechenden Stelle der Zeichenkette nach rechts. Je „Comparison Value“ kann „*“ nur einmal verwendet werden. Folglich entspricht „C*1“ „C1“, „CP1“ und „Chiller1“. Bei Verwendung von „Comparison Value“ zur Auswahl von „Object Identifiern“ werden diese als 32-Bit-Unsigned-Integer angesehen, das wichtigste Oktett ist das erste Oktett des „Object Identifier“. Bei Verwendung von Comparison Value zur Auswahl von CharacterStrings wird beim Vergleich die Groß- und Kleinschreibung berücksichtigt. Die Untersuchung von CharacterStrings auf Unterschiede (<, >, <=, >=) erfolgt basierend auf der numerischen Codierung jedes Zeichens von links nach rechts, wie in einem Telefonbuch. Beispiel: „ABC“ < „ABCD“ < „DE“ < „DEF“ Der Vergleich von Eigenschaften des Typs Date oder Time auf Unterschiede erfolgt chronologisch.

Beispiel:

1-Jan-1991 < 2-Feb-1992
7:00:00.00 < 23:00:00.00

Für Unterkomponenten von Date oder Time, z. B. das Jahr, können Platzhalter angegeben werden, wenn der Wert der Unterkomponente der Sonderwert „unspecified“ ist. Unterkomponenten von Date oder Time mit diesem Wert werden beim Vergleich übersprungen und die Unterkomponente gilt als gleich. Der Vergleichswert, der Wert der zu vergleichenden Eigenschaft oder beide können über Unterkomponenten verfügen, deren Wert(e) „unspecified“ sind. In jedem Fall gelten die Regeln für Übereinstimmungen nicht angegebener Werte.

Beispiel:

Tabelle 72: Vergleichswerte zur Auswahl von Eigenschaften

Vergleichswert	Eigenschaftswert	Grund
- Sun	= 29-Jan-1995 Sun	Sun = Sun
1-Jan-*	< 2-Feb-1995	1-Jan < 2-Feb
-Aug-	> 10-Mar-1995	Aug > Mar
*:59:00.00	> 6:23:99.0	59:00.00>23:00.00
6:*.*. *	=6:25:00.00	6 = 6

* nicht angegeben

Der Vergleich von Eigenschaften mit den Datentypen NULL, BOOLEAN, BIT STRING, OCTET STRING, ENUMERATED oder einer beliebigen komplexen Struktur einfacher Datentypen kann nur gleich oder ungleich (=, ≠) als Ergebnis haben.

8.12 Count

Dienste	Referenziert durch ReadRange im Parameter Range
Beschreibung	Der absolute Wert des Parameters Count gibt die Anzahl der auszulesenden Einträge an.

In der Parametergruppe „By Position“:

Ist Count positiv, ist der durch „Reference Index“ angegebene Eintrag der erste und älteste Eintrag, der ausgelesen und zurückgegeben wird.

Ist Count negativ, ist der durch „Reference Index“ angegebene Eintrag der letzte und neueste Eintrag. Count darf nicht Null sein.

In den Parametergruppen „By Sequence Number“ und „By Time“:

Ist Count positiv, ist der erste Eintrag mit einem neueren Zeitstempel als in Reference_Time angegeben der erste und somit älteste Eintrag, der ausgelesen und zurückgegeben wird.

Ist Count negativ, ist der neueste Eintrag mit einem Zeitstempel älter als der in Reference_Time angegebene letzte und neueste Eintrag. Count darf nicht Null sein.

8.13 COV Increment

Dienste	SubscribeCOVProperty
Beschreibung	Dieser Parameter des Typs REAL gibt die mindestens erforderliche Veränderung der überwachten Eigenschaft an, bei der eine COVNotification an hierfür konfigurierte COV-Clients gesendet wird. Dieser Parameter wird ignoriert, wenn der Datentyp der überwachten Eigenschaft nicht REAL ist. Ist die überwachte Eigenschaft ein Present_Value mit Datentyp REAL und ist der Parameter nicht vorhanden und das überwachte Objekt verfügt über die Eigenschaft COV_Increment, wird das zu verwendende „COV Increment“ aus der Eigenschaft COV_Increment des überwachten Objektes übernommen. Andernfalls ist das „COV Increment“ lokal unterschiedlich. Dies ermöglicht Abonnenten die Verwendung eines bereits festgelegten „COV Increment“ eines anderen Abonnements oder der Eigenschaft COV_Increment des überwachten Objektes.

8.14 Device Identifier

Dienste I-Have

Beschreibung Der Parameter „Device Identifier“ des Typs BACnetObjectIdentifier ist der Object_Identifier des Gerätes, das die I-Have-Dienstanfrage initiiert.

8.15 Device Instance Range High Limit

Dienste Who-Has, Who-Is

Beschreibung Dieser Parameter ist ein Unsigned-Integer im Bereich 0 bis 4.194.303. In Verbindung mit dem Parameter „Device Instance Range Low Limit“ definiert er die Geräte, die auf eine I-Have-Dienstanfrage (Who-Has-Dienst) bzw. I-Am-Dienstanfrage (Who-Is-Dienst) antworten dürfen, falls die Kriterien Object_Identifier oder Object_Name wie in Kapitel „Object Identifier“ und Kapitel „Object Name“ beschrieben erfüllt werden. Ist der Parameter „Device Instance Range High Limit“ vorhanden, muss der Parameter „Device Instance Range Low Limit“ ebenso vorhanden sein. Nur Geräte, deren Object_Identifier-Instanznummer im Bereich „Device Instance Range Low Limit $\leq$ Object_Identifier Instance Number $\leq$ Device Instance Range High Limit“ liegt, dürfen antworten. Der Wert von „Device Instance Range High Limit“ muss größer oder gleich dem Wert von „Device Instance Range Low Limit“ sein. Sind die Parameter „Device Instance Range Low Limit“ und „Device Instance Range High Limit“ nicht vorhanden, dürfen alle Geräte, die diese Meldung erhalten, mit einer I-Have-, bzw. I-Am-Dienstanfrage antworten.

8.16 Device Instance Range Low Limit

Dienste Who-Has, Who-Is

Beschreibung Dieser Parameter ist ein Unsigned-Integer im Bereich 0 bis 4.194.303. In Verbindung mit dem Parameter „Device Instance Range High Limit“ definiert er die Geräte, die auf eine I-Have- bzw. I-Am-Dienstanfrage antworten dürfen, falls die Kriterien „Object Identifier“ oder „Object Name“ wie in Kapitel „Object Identifier“ und Kapitel „Object Name“ beschrieben erfüllt werden. Ist der Parameter „Device Instance Range Low Limit“ vorhanden, muss der Parameter „Device Instance Range High Limit“ ebenso vorhanden sein und nur Geräte, deren Object_Identifier-Instanznummer im Bereich „Device Instance Range Low Limit $\leq$ Object_Identifier Instance Number $\leq$ Device Instance Range High Limit“ liegt, dürfen antworten. Der Wert von „Device Instance Range Low Limit“ muss größer oder gleich dem Wert von „Device Instance Range High Limit“ sein. Sind die Parameter „Device Instance Range Low Limit“ und „Device Instance Range High Limit“ nicht vorhanden, dürfen alle Geräte, die diese Meldung erhalten, mit einer I-Have- bzw. I-Am-Dienstanfrage antworten.

8.17 Enable/Disable

Dienste DeviceCommunicationControl

Beschreibung Dieser Parameter ist eine Enumeration, welche die Werte „ENABLE“, „DISABLE“ oder „DISABLE_INITIATION“ annehmen kann. Er gibt an, ob der Antwortende die gesamte Kommunikation der Netzwerkschnittstelle aktivieren oder deaktivieren oder die Initiierung deaktivieren kann. Hat dieser Parameter den Wert „ENABLE“, ist die Kommunikation aktiviert. Hat der Parameter den Wert „DISABLE“, ist die gesamte Kommunikation deaktiviert. Bei „DISABLE_INITIATION“ ist die Initiierung einer Kommunikation deaktiviert. Ist die Netzwerkkommunikation vollständig deaktiviert, können nur die APDUs DeviceCommunicationControl und ReinitializeDevice verarbeitet werden und es werden keine Meldungen initiiert. Ist die Initiierung einer Kommunikation deaktiviert, werden alle APDUs verarbeitet und ggf. Antworten zurückgegeben. Es werden jedoch keine Meldungen initiiert, außer bei I-Am-Anfragen, die als Antwort auf Who-Is-Anfragen initiiert werden. In diesem Zustand sendet ein Gerät, das I-Am-Anfragen unterstützt, eine I-Am-Anfrage für jede empfangene Who-Is-Anfrage, jedoch nur, wenn die Who-Is-Anfrage keinen Adressbereich enthält oder das Gerät im Adressbereich enthalten ist.

8.18 End Of File

Dienste AtomicReadFile

Beschreibung Der Parameter „End Of File“ des Typs BOOLEAN ist „TRUE“, falls diese Antwort das letzte Oktett der Datei enthält, andernfalls ist der Parameter „FALSE“. Der Parameter wird für die Überprüfung des Dateiendes verwendet, da die Anzahl der zurückgegebenen Oktette aufgrund in der Datei verbleibender Daten geringer als „Requested Octet Count“ oder „Returned Record Count“ geringer als „Requested Octet Count“ sein könnte. Dieser Parameter bietet auch eine von den Daten unabhängige Methode zur Bestimmung des Dateiendes für den Client-User.

8.19 Enrollment Filter

Dienste	GetEnrollmentSummary
Beschreibung	Dieser Parameter des Typs BACnetRecipientProcess bietet eine Möglichkeit zur Einschränkung der Objekte, die zusammengefasst werden sollen. Nur Objekte, die zu bestätigende oder nicht zu bestätigende Benachrichtigungen über BACnetRecipient und „Process Identifier“ erhalten, werden aufgeführt. In diesem Fall bedeutet „Benachrichtigungen erhalten“, dass ein Objekt, welches ein Ereignis auslöst, auf ein Notification-Class-Objekt mit einem oder mehreren BACnetDestinations verweist, die den angegebenen „Process Identifier“ und BACnetRecipient enthalten. Ist dieser Parameter nicht vorhanden, bedeutet dies, dass ereignisauslösende Objekte ohne Berücksichtigung des Enrollment-Zustandes zusammengefasst werden.

8.20 Error Type

Dienste	AcknowledgeAlarm, AtomicReadFile, AtomicWriteFile, ConfirmedCOVNotification, ConfirmedEventNotification, DeviceCommunicationControl, GetAlarmSummary, GetEnrollmentSummary, GetEventInformation, LifeSafetyOperation, Object-Access-Dienste, ReadProperty, ReadPropertyConditional, ReadPropertyMultiple, ReadRange, ReinitializeDevice, RemoveListElement, SubscribeCOV, SubscribeCOVProperty, VT-Close, VT-Data, VT-Open, WriteProperty, WritePropertyMultiple
Beschreibung	Dieser Parameter besteht aus zwei Komponenten: (1) der Fehlerklasse und (2) dem Fehlercode siehe Kapitel „Fehler-, Ablehnungs- und Abbruchcodes“. Die in bestimmten Situationen zurückgegebene Fehlerklasse und der Fehlercode sind nachfolgend aufgeführt:

In Dienst CreateObject:

Tabelle 73: Fehlerklasse und Fehlercode des Dienstes CreateObject

Situation	Fehlerklasse	Fehlercode
Das Gerät kann den für das neue Objekt erforderlichen Speicher nicht zuweisen.	RESOURCES	NO_SPACE_FOR_OBJECT
Das Gerät unterstützt die Erstellung dieses Objektes aus einem anderen Grund als zu geringem Speicherplatz nicht.	OBJECT	DYNAMIC_CREATION_NOT_SUPPORTED
Das zu erstellende Objekt ist bereits vorhanden.	OBJECT	OBJECT_IDENTIFIER_ALREADY_EXISTS
Der Datentyp einer in der List_Of_Initial_Values angegebenen Eigenschaft entspricht nicht dem Datentyp der durch den Property_Identifier referenzierten Eigenschaft.	PROPERTY	INVALID_DATATYPE
Ein in der List_Of_Initial_Values angegebener Wert liegt außerhalb des Wertebereiches, der durch den Property_Identifier angegebenen Eigenschaft.	PROPERTY	VALUE_OUT_OF_RANGE
In der List_Of_Initial_Values wurde ein Property_Identifier angegeben, der Objekten des zu erstellenden Datentyps nicht bekannt ist.	PROPERTY	UNKNOWN_PROPERTY
Ein Wert des Typs CharacterString in der List_Of_Initial_Values entspricht keinem unterstützten Zeichensatz.	PROPERTY	CHARACTER_SET_NOT_SUPPORTED
Eine durch den Property_Identifier in der List_Of_Initial_Values angegebene Eigenschaft unterstützt die Initialisierung mit dem Dienst CreateObject nicht.	PROPERTY	WRITE_ACCESS_DENIED

In Dienst DeleteObject:

Tabelle 74: Fehlerklasse und Fehlercode des Dienstes DeleteObject

Situation	Fehlerklasse	Fehlercode
Das zu löschende Objekt ist nicht vorhanden.	OBJECT	UNKNOWN_OBJECT
Das Objekt ist vorhanden aber kann nicht gelöscht werden.	OBJECT	OBJECT_DELETION_NOT_PERMITTED

In Dienst ReadProperty, ReadPropertyMultiple:

Tabelle 75: Fehlerklasse und Fehlercode der Dienste DeleteObject
ReadProperty/ReadPropertyMultiple

Situation	Fehlerklasse	Spaltenüberschrift
Das angegebene Objekt ist nicht vorhanden.	OBJECT	UNKNOWN_OBJECT
Die angegebene Eigenschaft ist nicht vorhanden.	PROPERTY	UNKNOWN_PROPERTY
Es ist ein Feldindex angegeben, die Eigenschaft ist jedoch kein Feld.	PROPERTY	PROPERTY_IS_NOT_AN_ARRAY
Es ist ein Feldindex außerhalb des in der Eigenschaft vorhandenen Bereiches angegeben.	PROPERTY	INVALID_ARRAY_INDEX

In Dienst WriteProperty, WritePropertyMultiple:

Tabelle 76: Fehlerklasse und Fehlercode der Dienste WriteProperty/WritePropertyMultiple

Situation	Fehlerklasse	Fehlercode
Das angegebene Objekt ist nicht vorhanden.	OBJECT	UNKNOWN_OBJECT
Die angegebene Eigenschaft ist nicht vorhanden.	PROPERTY	UNKNOWN_PROPERTY
Es ist ein Feldindex angegeben, die Eigenschaft ist jedoch kein Feld.	PROPERTY	PROPERTY_IS_NOT_AN_ARRAY
Es ist ein Feldindex außerhalb des in der Eigenschaft vorhandenen Bereiches angegeben.	PROPERTY	INVALID_ARRAY_INDEX
Die angegebene Eigenschaft ist derzeit für den Dienst WritePropertyMultiple nur lesbar und der Anforderer kann nicht mit dem Dienst WriteProperty in die Eigenschaft schreiben.	PROPERTY	WRITE_ACCESS_DENIED
Der Datentyp des angegebenen Wertes ist für die angegebene Eigenschaft ungültig.	PROPERTY	INVALID_DATATYPE
Die Eigenschaft ist Object_Name und der Name wird in diesem Gerät bereits verwendet.	PROPERTY	DUPLICATE_NAME
Die Eigenschaft ist Object_Identifier und der Identifier wird in diesem Gerät bereits verwendet.	PROPERTY	DUPLICATE_OBJECT_ID
Der angegebene Wert liegt außerhalb des für diese Eigenschaft möglichen Wertebereiches.	PROPERTY	VALUE_OUT_OF_RANGE
Es ist nicht genügend Speicherplatz für das Speichern des neuen Wertes vorhanden.	RESOURCES	NO_SPACE_TO_WRITE_PROPERTY

8.21 Event Enable

Dienste GetEventInformation

Beschreibung Dieser Parameter des Typs BACnetEventTransitionBits gibt den Wert der Eigenschaft Event_Enable des Objektes an.

8.22 Event Object Identifier

Dienste AcknowledgeAlarm, ConfirmedEventNotification, UnconfirmedEventNotification

Beschreibung **In Dienst AcknowledgeAlarm:**
Dieser Parameter des Typs BACnetObjectIdentifier gibt den Parameter „Event Object Identifier“ der Ereignisbenachrichtigung an, den diese Bestätigung beantwortet. Dies ist das Objekt, das die Ereignisbenachrichtigung, die beantwortet wird, initiiert hat.

In Dienst ConfirmedEventNotification, UnconfirmedEventNotification:
Dieser Parameter des Typs BACnetObjectIdentifier gibt den Object_Identifier des Objektes an, das die Benachrichtigung initiiert. Der Parameter wird vom Dienst AcknowledgeAlarm verwendet, um das Objekt zu bestimmen, dessen Benachrichtigung bestätigt wird.

8.23 Event Priorities

Dienste GetEventInformation

Beschreibung Dieser Parameter des Typs „BACnetARRAY [3] of Unsigned“ gibt die in der Eigenschaft Priority angegebenen Prioritäten des entsprechenden Notification-Class-Objektes an.

8.24 Event State

Dienste GetEnrollmentSummary, GetEventInformation

Beschreibung Dieser Parameter des Typs BACnetEventState gibt den Wert der Eigenschaft Event_State des Objektes an.

8.25 Event State Acknowledged

Dienste AcknowledgeAlarm

Beschreibung Dieser Parameter des Typs BACnetEventState entspricht dem Wert „To State“ der zu bestätigenden Ereignisbenachrichtigung. Der Parameter ist vorhanden, damit das entfernte Gerät, das die Ereignisbenachrichtigung initiiert hat, sicherstellen kann, dass der bestätigte Zustand in der Eigenschaft Acked_Transitions des initiiierenden Objektes gespeichert wird.

8.26 Event State Filter

Dienste GetEnrollmentSummary

Beschreibung Dieser Parameter bietet eine Möglichkeit zur Einschränkung der Objekte, die Ereignisse auslösen und in der Zusammenfassung enthalten sind. Der Parameter kann folgende Werte annehmen:

{OFFNORMAL, FAULT, NORMAL, ALL, ACTIVE}

Es sind nur ereignisauslösende Objekte enthalten, deren Eigenschaft Event_State mit dem in diesem Parameter angegebenen Wert übereinstimmt. Ist der Wert „ALL“ angegeben, werden alle ereignisauslösenden Objekte zusammengefasst, unabhängig vom Wert der Eigenschaft Event_State. Ist der Wert „ACTIVE“ angegeben, werden nur ereignisauslösende Objekte zusammengefasst, deren Eigenschaft Event_State nicht den Wert „NORMAL“ hat. Ist dieser Parameter nicht vorhanden, wird der Wert „ALL“ angenommen.

8.27 Event Time Stamps

Dienste GetEventInformation

Beschreibung Dieser Parameter des Typs „BACnetARRAY [3] of BACnetTimeStamp“ gibt die Zeitstempel der letzten Ereignisbenachrichtigungen für TO-OFFNORMAL-, TO-FAULT- und TO-NORMAL-Ereignisse an.

8.28 Event Type

Dienste	ConfirmedEventNotification, GetEnrollmentSummary, UnconfirmedEventNotification
Beschreibung	Dieser Parameter des Typs BACnetEventType gibt den Typ des stattgefundenen Ereignisses an.

8.29 Event Type Filter

Dienste	GetEnrollmentSummary
Beschreibung	Dieser Parameter bietet eine Möglichkeit, nur die ereignisauslösenden Objekte zusammenzufassen, die Ereignisbenachrichtigungen des in diesem Parameter angegebenen Event_Type generieren können.

8.30 Event Values

Dienste	ConfirmedEventNotification, UnconfirmedEventNotification
Beschreibung	Dieser Parameter des Typs BACnetNotificationParameters gibt Werte an, die für das entsprechende Ereignis relevant sind und deren Inhalt vom Typ des Objektes abhängt, das die Benachrichtigung initiiert hat, siehe Tabellen in Kapitel „Objektinternes Melden“. Dieser Parameter darf nur vorhanden sein, wenn der Parameter Notify Type EVENT oder ALARM ist. Die zurückgegebenen Werte sind entweder eine Liste mit Eigenschaftswerten, die Eigenschaften des im „Event Object Identifier“ angegebenen Objektes darstellt oder einer der in Tabelle „Benachrichtigungsparameter für Standard-Ereignistypen“ (Kapitel „Objektinternes Melden“) definierten Standard-Benachrichtigungsparameter für Ereignistypen.

8.31 File Data

Dienste	AtomicReadFile, AtomicWriteFile
Beschreibung	Dieser Parameter besteht aus einem OCTET STRING, der die angefragten Dateidaten (AtomicReadFile) oder die zu schreibenden Dateidaten (AtomicWriteFile) enthält.

8.32 File Identifier

Dienste AtomicReadFile, AtomicWriteFile

Beschreibung Dieser Parameter ist der Object_Identifier des File-Objektes, der die zu schreibende/lesende Datei bestimmt.

8.33 File Record Data

Dienste AtomicReadFile, AtomicWriteFile

Beschreibung Dieser Parameter besteht aus einer „List of OCTET STRINGS“ mit den angefragten/zu schreibenden Daten.

8.34 File Start Position

Dienste AtomicReadFile, AtomicWriteFile

Beschreibung Dieser Parameter des Typs INTEGER gibt die Oktettanzahl ab Dateianfang an, ab der die Daten geschrieben werden. Eine „File Start Position“ von „0“ ist das erste Oktett der Datei. Eine „File Start Position“ von „-1“ gibt das Ende der aktuellen Datei an, d. h. die Datei wird erweitert.

8.35 File Start Record

Dienste AtomicReadFile, AtomicWriteFile

Beschreibung Dieser Parameter des Typs INTEGER gibt die Eintragsanzahl ab Dateianfang an, ab der die Daten gelesen (AtomicReadFile) oder geschrieben (AtomicWriteFile) wurden. Ein „File Start Record“ von „0“ ist der erste Eintrag der Datei. Ein „File Start Record“ von „-1“ gibt das Ende der aktuellen Datei an, d. h. die Datei wird erweitert.

8.36 First Failed Element Number

Dienste CreateObject, AddListElement, RemoveListElement

Beschreibung **In Dienst AddListElement, RemoveListElement:**
Dieser Parameter des Typs Unsigned gibt die numerische Position (beginnend mit 1) des fehlerhaften Elementes im Parameter „List of Elements“ der Anfrage an. Gilt die Anfrage aus einem anderen Grund als dem Parameter „List of Elements“ als ungültig, ist „First Failed Element Number“ gleich „Null“.

In Dienst CreateObject:
Dieser Parameter des Typs Unsigned gibt die numerischen Werte (beginnend mit „1“) des fehlerhaften „Initial Value“ im Parameter „List of Initial Values“ der Anfrage an. Gilt die Anfrage aus einem anderen Grund als dem Parameter „List of Initial Values“ als ungültig, ist „First Failed Element Number“ „Null“.

8.37 First Failed Write Attempt

Dienste WritePropertyMultiple

Beschreibung Dieser Parameter des Typs BACnetObjectPropertyReference gibt „Object Identifier“, „Property Identifier“ und „Property Array Index“ des ersten fehlgeschlagenen Elementes der „List of Write Access Specification“ an, bei dem der Schreibvorgang fehlgeschlagen ist.

8.38 First Sequence Number

Dienste ReadRange

Beschreibung Dieser Parameter des Typs Unsigned32 gibt die Sequenznummer des ersten zurückgegebenen Elementes an. Dieser Parameter ist nur vorhanden, wenn der Parameter Range der Anfrage den Typ „By Sequence Number“ oder „By Time“ hat und der „Item Count“ größer „0“ ist.

8.39 From State

Dienste	ConfirmedEventNotification, UnconfirmedEventNotification
Beschreibung	Dieser Parameter des Typs BACnetEventState gibt den Wert der Eigenschaft Event_State des Objektes vor Auftreten des Ereignisses an, das diese Benachrichtigung initiiert hat. Dieser Parameter darf nur vorhanden sein, wenn der Parameter „Notify Type“ EVENT oder ALARM ist.

8.40 I-Am Device Identifier

Dienste	I-Am
Beschreibung	Der Parameter „I-Am Device Identifier“ des Typs BACnetObjectIdentifier ist der Device-Object_Identifier des Gerätes, das die I-Am-Dienstanfrage initiiert.

8.41 Initiating Device Identifier

Dienste	ConfirmedCOVNotification, ConfirmedEventNotification, UnconfirmedCOVNotification, UnconfirmedEventNotification
Beschreibung	Dieser Parameter des Typs BACnetObjectIdentifier ist der Device-Object_Identifier des Gerätes, das eine der obenstehenden Dienstanfragen initiiert.

8.42 Issue Confirmed Notifications

Dienste SubscribeCOV, SubscribeCOVProperty

Beschreibung Dieser Parameter des Typs BOOLEAN gibt an, ob das COV-Servergerät bei Änderungen ConfirmedCOVNotifications („TRUE“) oder UnconfirmedCOVNotifications („FALSE“) ausgibt. Falls vorhanden gibt dieser Parameter an, ob ein Abonnement oder Neu-Abonnement erfolgt und ob die Lebensdauer auf den ursprünglichen Zustand zurückgesetzt wird. Ist keiner der Parameter Lifetime oder „Issue Confirmed Notifications“ vorhanden, handelt es sich um eine Stornierungsanfrage. Ist der Parameter Lifetime vorhanden, muss der Parameter „Issue Confirmed Notifications“ ebenfalls vorhanden sein.

8.43 Item Count

Dienste ReadRange

Beschreibung Dieser Parameter des Typs Unsigned gibt die Anzahl der zurückgegebenen Elemente an.

8.44 Item Data

Dienste ReadRange

Beschreibung Dieser Parameter des Typs Unsigned gibt die Anzahl der zurückgegebenen Elemente an.

8.45 Last Received Object Identifier

Dienste GetEventInformation

Beschreibung Dieser optionale Parameter des Typs BACnetObjectIdentifier gibt den letzten in einer vorhergehenden GetEventInformation-ACK-Meldung empfangenen „Object Identifier“ an, falls dessen Parameter „More Events“ gleich „TRUE“ war. Ist dieser Parameter nicht vorhanden, beginnt die Rückgabe mit dem ersten Objekt, das die Kriterien für „Active Event States“ erfüllt. Es wird eine feste Reihenfolge für die Bearbeitung angenommen. Die tatsächliche Reihenfolge ist lokal unterschiedlich. Ist der „Last Received Object Identifier“ des antwortenden Gerätes bereits ungültig (z. B. wenn das Objekt nicht mehr vorhanden ist), wird der Dienst fortgesetzt, falls das Objekt ermittelt werden kann, das nach dem nicht mehr vorhandenen Objekt zu bearbeiten ist. Andernfalls wird Result(-) mit der Fehlerklasse OBJECT und dem Fehlercode UNKNOWN_OBJECT zurückgegeben.

8.46 Lifetime

Dienste SubscribeCOV, SubscribeCOVProperty

Beschreibung Dieser Parameter des Typs Unsigned gibt die gewünschte Lebensdauer für ein Abonnement in Sekunden an. Ein Wert von „Null“ gibt an, dass das Abonnement unbegrenzt gültig ist und keine automatische Stornierung erfolgt. Jeder andere Wert gibt die Zeit in Sekunden vor der automatischen Stornierung des Abonnements an. Ist keiner der Parameter Lifetime oder „Issue Confirmed Notifications“ vorhanden, handelt es sich um eine Stornierungsanfrage. Ist der Parameter Lifetime vorhanden, muss der Parameter „Issue Confirmed Notifications“ ebenfalls vorhanden sein.

8.47 List of Alarm Summaries

Dienste GetAlarmSummary

Beschreibung Die „List of Alarm Summaries“ enthält Null oder mehr Alarm-Zusammenfassungen. Jede Alarm-Zusammenfassung besteht aus drei Parameter: „Object Identifier“, „Alarm State“ und „Acknowledged States“. Ist die Länge der Liste „Null“, liegen für dieses Gerät keine aktiven Alarmer vor.

8.48 List of Elements

Dienste Object Access, RemoveListElement

Beschreibung Dieser Parameter gibt eines oder mehrere Elemente an, die der im Parameter „Property Identifier“ angegebenen Eigenschaft hinzugefügt werden. Der Datentyp der Parameterelemente wird durch den Objekttyp im Parameter „Object Identifier“ bestimmt.

8.49 List of Enrollment Summaries

Dienste GetEnrollmentSummary

Beschreibung Die „List of Enrollment Summaries“ enthält Null oder mehr Enrollment-Zusammenfassungen. Jede Enrollment-Zusammenfassung besteht aus bis zu fünf Parametern: „Object Identifier“, „Event Type“, „Event State“, Priority und optional Notification-Class. Ist die Länge der Liste Null, liegen keine ereignisauslösenden Objekte vor, die den in der Anfrage angegebenen Suchkriterien entsprechen.

8.50 List of Event Summaries

Dienste GetEventInformation

Beschreibung Die „List of Event Summaries“ enthält Null oder mehr Ereigniszusammenfassungen. Jede Ereignis-Zusammenfassung besteht aus sieben Parametern: „Object Identifier“, „Event State“, „Acknowledged Transitions“, „Event Time Stamps“, „Notify Type“, „Event Enable“ und „Event Priorities“. Ist die Länge der Liste Null, liegen für dieses Gerät keine ereignisauslösenden Objekte mit dem Ereigniszustand ACTIVE vor.

8.51 List of Properties

Dienste	Referenziert durch den Dienst WritePropertyMultiple im Parameter „Write Access Specification“
Beschreibung	Dieser Parameter gibt eine Liste mit einer oder mehreren Eigenschaften des jeweiligen Objektes, des in jede Eigenschaft zu schreibenden Wertes und der dem Schreibvorgang zugewiesenen Priorität an. Jedes Element der Liste enthält folgende Parameter: • Property Identifier • Property Array Index • Property Value • Priority (siehe Kapitel „Gesamtliste der BACnet-Dienst-Parameter“)

8.52 List of Property References

Dienste	ReadPropertyConditional, referenziert durch den Dienst ReadPropertyMultiple im Parameter „Read Access Specification“
Beschreibung	In Dienst ReadPropertyConditional: Falls vorhanden gibt dieser optionale Parameter, falls eine Liste mit einer oder mehreren BACnetPropertyReferences an, wobei jede Reference direkt auf eine bestimmte Eigenschaft der auf Grundlage der „Object Selection Criteria“ ausgewählten Objekte verweist. Der Wert „ALL“ gibt an, dass alle Eigenschaften ausgewählter Objekte zurückgegeben werden, einschließlich proprietärer Eigenschaften. Der Wert „REQUIRED“ gibt an, dass nur Eigenschaften zurückgegeben werden, die mit Code „R“ oder „W“ konform sind. Der „Property Identifier“ OPTIONAL gibt an, dass nur Eigenschaften zurückgegeben werden, die mit Code „O“ konform sind. Siehe Spezifikationen für das entsprechenden Objekt in Kapitel „Objekte im nicht-nativen Betrieb“. Ist der Parameter nicht vorhanden, werden im Kapitel „Gesamtliste der BACnet-Dienst-Parameter“, Parameter „List of Read Access Results“ der Result(+)-Antwort keine Eigenschaften zurückgegeben, der „Object Identifier“ wird jedoch im entsprechenden Parameter zurückgegeben.

In Dienst ReadPropertyMultiple (im Parameter „Read Access Specification“):

Dieser Parameter ist eine Liste einer oder mehrerer BACnetPropertyReferences, die jeweils direkt einer angegebenen Eigenschaft des oben bestimmten Objektes entspricht. Der „Property Identifier“ ALL gibt an, dass auf alle angegebenen Eigenschaften des Objektes zugegriffen wird, einschließlich proprietärer Eigenschaften. Der „Property Identifier“ REQUIRED gibt an, dass nur Eigenschaften zurückgegeben werden, die mit Code „R“ oder „W“ konform sind. Der „Property Identifier“ OPTIONAL gibt an, dass nur Eigenschaften zurückgegeben werden, die mit Code „O“ konform sind (siehe Spezifikationen für das entsprechende Objekt in Kapitel „Objekte im nicht-nativen Betrieb“).

8.53 List of Read Access Results

Dienste ReadPropertyConditional, ReadPropertyMultiple

Beschreibung **In Dienst ReadPropertyConditional:**

Der Parameter „List of Read Access Results“ gibt für jede angegebene Eigenschaft der auf Grundlage der „Object Selection Criteria“ ausgewählten Objekte an, ob das Auslesen der Eigenschaft erfolgreich war oder nicht. Jedes „Read Access Result“ enthält „Object Identifier“, „Property Identifier“ und unter Umständen „Property Array Index“ für jeden Leseversuch. War der Leseversuch erfolgreich, ist der nächste Parameter der Wert der angegebenen Eigenschaft. Wurde die Eigenschaft nicht korrekt ausgelesen, ist der nächste Parameter ein Fehlercode, der den Grund für das Fehlschlagen angibt. Sind keine Objekte auf Grundlage der „Object Selection Criteria“ ausgewählt, hat die „List of Read Access Results“ die Länge Null. War die „List of Property References“ in der Anfrage nicht vorhanden, besteht die „List of Read Access Results“ aus einer „List of Read Access Results“ mit leeren Eigenschaftslisten und gibt so die „Object Identifier“ für Objekte an, die der „Selection Criteria“ entsprechen.

In Dienst ReadPropertyMultiple:

Der Parameter „List of Read Access Results“ gibt den Erfolg oder das Fehlschlagen eines Zugriffs auf jede angegebene Eigenschaft an. Der Inhalt der „Read Access Results“ ist in Kapitel „Parameter des Dienstes ReadPropertyMultiple“ > „Read Access Result“ beschrieben.

8.54 List of Read Access Specifications

Dienste	ReadPropertyMultiple
Beschreibung	Dieser Parameter besteht aus einer Liste mit einer oder mehreren „Read Access Specifications“. Jede Spezifikation besteht aus zwei Parametern: einem „Object Identifier“ und einer „List of Property References“, siehe Kapitel „Parameter des Dienstes ReadPropertyMultiple“ > „Read Access Specification“.

8.55 List of Results

Dienste	Referenziert durch ReadPropertyConditional im Parameter „List of Read Access Results“
Beschreibung	Das Ergebnis des Auslesens einer Eigenschaft ist entweder der Present_Value der Eigenschaft oder ein Fehlercode, der den Grund für das Fehlschlagen des Zugriffversuchs angibt. Jedes Element einer „List of Results“ enthält einen „Property Identifier“ und unter Umständen einen „Property Array Index“, auf den entweder ein „Property Value“ oder ein „Property Access Error“ folgt.

8.56 List of Selection Criteria

Dienste	Referenziert durch ReadPropertyConditional im Parameter „Object Selection Criteria“
Beschreibung	Falls vorhanden besteht dieser Parameter aus einer Liste mit einer oder mehreren Eigenschaftsbeziehungen, die für die Auswahl bestimmter Objekte des oben angegebenen Typs verwendet wird. Jede Beziehung besteht aus „Property Identifier“, „Relation Specifier“, „Comparison Value“ und unter Umständen „Property Array Index“. Zusammen bildet jede Parametergruppe einen booleschen Ausdruck, der bei Auswertung durch den antwortenden BACnet-User „TRUE“ oder „FALSE“ ist. Ist mehr als eine Eigenschaftsbeziehung angegeben, werden die Ergebnisse aller Auswertungen gemäß dem Parameter „Selection Logic“ kombiniert, um einen Gesamtwert zu erhalten, der „TRUE“ oder „FALSE“ ist. Bei „TRUE“ gibt der Antwortende Werte der für dieses Objekt in der „List of Property References“ angegebenen Eigenschaften zurück.

8.57 List of Values

Dienste	ConfirmedCOVNotification, UnconfirmedCOVNotification
Beschreibung	Dieser Parameter gibt eine Liste mit einem oder mehreren Eigenschaftswerten an, deren Inhalt vom überwachten Objekt abhängt. Die Tabelle in Kapitel „Melden bei Wertänderungen (Change Of Value / State Reporting)“ fasst die BACnet-Standardobjekte und Eigenschaftswerte zusammen, die im Parameter „List of Values“ zurückgegeben werden, wenn für die Objekte COV-Reporting aktiviert ist. Die zurückgegebenen Eigenschaftswerte sind in der genannten Tabelle angegeben.

8.58 List of VT Session Identifiers

Dienste	VT-Close
Beschreibung	Gibt der Parameter „Error Type“ den Fehlercode VT_SESSION_TERMINATION_FAILURE zurück, muss dieser Parameter vorhanden sein. Gibt der Parameter Fehlercode einen anderen Fehler zurück, darf dieser Parameter nicht vorhanden sein. Der Parameter „List of VT Session Identifiers“ besteht aus einer Liste mit mindestens einem „VT Session Identifier“. Jeder „VT Session Identifier“ gibt eine bestimmte VT-Session an, die nicht beendet werden konnte. Die zurückgegebenen „Session Identifier“ sind lokal in Bezug auf das Gerät, das die Result(-)-Anfrage empfängt, den anfragenden VT-User.

8.59 List of Write Access Specifications

Dienste	WritePropertyMultiple
Beschreibung	Jede „Write Access Specification“ gibt Informationen an, die für die Änderung einer oder mehrerer Eigenschaften eines BACnet-Objektes erforderlich sind. Die Spezifikation besteht aus bis zu fünf Parametern: „Object Identifier“, einer Liste von Properties mit jeweils einem „Property Identifier“, optionalem „Property Array Index“, „Property Value“ und optionalem Priority.

8.60 List of Results

Dienste	Referenziert durch ReadPropertyConditional im Parameter „List of Read Access Results“
Beschreibung	Das Ergebnis des Auslesens einer Eigenschaft ist entweder der Wert der Eigenschaft oder ein Fehlercode, der den Grund für das Fehlschlagen des Zugriffversuchs angibt. Hierbei ist zu beachten, dass der Parameter „List of Results“ nicht vorhanden sein darf, wenn der Parameter „List of Property References“ nicht in der ursprünglichen Anfrage enthalten war. Dies bietet eine Möglichkeit zur Abfrage einer Liste von „Object Identifiers“ aller Objekte, welche die angegebenen „Object Selection Criteria“ erfüllen, ohne die Werte bestimmter Eigenschaften dieser Objekte zu erhalten.

8.61 Max APDU Length Accepted

Dienste	I-Am
Beschreibung	Diese Eigenschaft des Typs Unsigned gibt die maximale Anzahl der Oktette an, die in einer einzelnen, nicht teilbaren APDU enthalten sein dürfen. Der Wert dieses Parameters entspricht dem Wert der Eigenschaft Max_APDU_Length_Accepted des Device-Objektes.

8.62 Message Text

Dienste	ConfirmedEventNotification, UnconfirmedEventNotification
Beschreibung	Dieser optionale Parameter des Typs CharacterString gibt eine Zeichenkette druckbarer Zeichen an. Der Parameter kann für eine Nachricht verwendet werden, die je nach Ereignis gespeichert oder angezeigt wird. Der Inhalt der Nachricht ist lokal unterschiedlich.

8.63 Monitored Object Identifier

Dienste	ConfirmedCOVNotification, SubscribeCOV, SubscribeCOVProperty, UnconfirmedCOVNotification
Beschreibung	In Dienst SubscribeCOV, SubscribeCOVProperty: Dieser Parameter des Typs BACnetObjectIdentifier gibt den Identifier des Objektes im empfangenden Gerät an, der die Eigenschaft enthält, für die ein Abonnement gewünscht ist. In Dienst ConfirmedCOVNotification, UnconfirmedCOVNotification: Dieser Parameter des Typs BACnetObjectIdentifier gibt den Object_Identifier des geänderten Objektes an.

8.64 Monitored Property Identifier

Dienste	SubscribeCOVProperty
Beschreibung	Dieser Parameter des Typs BACnetPropertyReference gibt den „Property Identifier“ und optional den Feldindex an, für den ein Abonnement gewünscht ist. Wird COV-Reporting für eine Eigenschaft mit einem Array-Datentyp unterstützt, ist die Unterstützung von COV-Abonnements für alle oder nur für bestimmte Elemente des Feldes lokal unterschiedlich.

8.65 More Events

Dienste	GetEventInformation
Beschreibung	Dieser Parameter des Typs BOOLEAN gibt an, ob mehrere Objekte vorhanden sind („TRUE“), die den Active-Event-State-Kriterien der Dienstanfrage entsprechen, jedoch nicht in der Antwort zurückgegeben werden konnten, oder nicht („FALSE“).

8.66 Notification-Class

Dienste	ConfirmedEventNotification, GetEnrollmentSummary, UnconfirmedEventNotification
Beschreibung	In Dienst GetEnrollmentSummary: Dieser optionale Parameter des Typs Unsigned gibt die Klasse der Benachrichtigungen an, die das Objekt generiert und verweist eindeutig auf ein Notification-Class-Objekt, dessen Eigenschaft Notification_Class denselben Wert hat. In Dienst ConfirmedEventNotification, UnconfirmedEventNotification: Dieser Parameter des Typs Unsigned gibt die Benachrichtigungsklasse des Ereignisses an. Die Definition der verschiedenen Benachrichtigungsklassen ist lokal unterschiedlich, siehe Kapitel „Objektinternes Melden“, Kapitel „Auftreten und Benachrichtigung von Alarmen und Ereignissen“ und Kapitel „Notification-Class-Objekt“ für Informationen zum Notification-Class-Objekt.

8.67 Notification-Class Filter

Dienste	GetEnrollmentSummary
Beschreibung	Dieser Parameter des Typs Unsigned bietet eine Möglichkeit, nur die ereignisauslösenden Objekte zusammenzufassen, die Ereignisbenachrichtigungen mit einer in diesem Parameter angegebenen Notification-Class generieren können. Ist der Notification-Class Filter nicht vorhanden, bedeutet dies, dass alle ereignisauslösenden Objekte ohne Berücksichtigung der Notification-Class zusammengefasst werden.

8.68 Notify Type

Dienste	ConfirmedEventNotification, GetEventInformation, UnconfirmedEventNotification
Beschreibung	In Dienst UnconfirmedEventNotification, ConfirmedEventNotification: Dieser Parameter des Typs BACnetNotifyType gibt an, ob die Benachrichtigung ein Ereignis oder ein Alarm ist oder eine Antwort auf eine vorherige Benachrichtigung: {ALARM, EVENT, ACK_NOTIFICATION} In Dienst GetEventInformation: Dieser Parameter des Typs BACnetNotifyType gibt den Wert der Eigenschaft Notify_Type des Objektes an.

8.69 Object Identifier

Dienste AddListElement, GetAlarmSummary, GetEnrollmentSummary, GetEventInformation, I-Have, LifeSafetyOperation, ReadProperty, referenziert durch ReadPropertyConditional im Parameter „Read Access Result“, ReadRange, RemoveListElement, Who-Has, WriteProperty, referenziert durch WritePropertyMultiple im Parameter „Write Access Specification“, referenziert durch ReadPropertyMultiple im Parameter „Read Access Specification“

Beschreibung **In Dienst GetAlarmSummary:**
Dieser Parameter des Typs BACnetObjectIdentifier gibt das ereignisauslösende Objekt an, dessen Eigenschaft Event_State nicht NORMAL ist und dessen Eigenschaft Notify_Type den Wert „ALARM“ hat.

In Dienst GetEnrollmentSummary:
Dieser Parameter des Typs BACnetObjectIdentifier gibt ein Objekt an, das den Suchkriterien entspricht.

In Dienst GetEventInformation Service:
Dieser Parameter des Typs BACnetObjectIdentifier gibt ein ereignisauslösendes Objekt an, dessen Eigenschaft Event_State nicht den Wert „NORMAL“ hat oder bei dessen Eigenschaft Acked_Transitions mindestens eines der folgenden Bits den Zustand FAULT hat: TO-OFFNORMAL, TO-FAULT, TO-NORMAL.

In Dienst LifeSafetyOperation:
Dieser Parameter des Typs BACnetObjectIdentifier gibt das BACnet-Objekt an, an das Life-Safety-Requests weitergeleitet werden. Ist dieser Parameter nicht vorhanden, werden alle entsprechenden Objekte im empfangenden BACnet-Gerät stummgeschaltet oder zurückgesetzt, je nach Anfrage.

In Dienst AddListElement, RemoveListElement, WritePropertyMultiple:
Dieser Parameter des Typs BACnetObjectIdentifier stellt eine Möglichkeit zur Bestimmung des Objektes dar, dessen angegebene Listeneigenschaft von diesem Dienst bearbeitet werden soll.

In Dienst ReadProperty in Parametergruppe Argument:
Dieser Parameter des Typs BACnetObjectIdentifier bietet eine Möglichkeit zur Bestimmung des Objektes, dessen Eigenschaft ausgelesen und an den Client-BACnet-User zurückgegeben werden soll.

In Dienst ReadProperty in Parametergruppe Result(+):

Dieser Parameter des Typs BACnetObjectIdentifier gibt das Objekt an, dessen Eigenschaft ausgelesen wurde und an den Client-BACnet-User zurückgegeben wird.

In Dienst ReadPropertyMultiple im Parameter Read Access Specification:

Dieser Parameter des Typs BACnetObjectIdentifier bietet eine Möglichkeit zur Bestimmung des Objektes, dessen Eigenschaften ausgelesen und an den Sender der Dienstanfrage zurückgegeben werden.

In Dienst ReadPropertyConditional im Parameter „Read Access Result“:

Dieser Parameter des Typs BACnetObjectIdentifier gibt das Objekt an, dessen Eigenschaften an den Sender der Dienstanfrage zurückgegeben werden.

In Dienst ReadRange in Parametergruppe Argument:

Dieser Parameter des Typs BACnetObjectIdentifier gibt das auszulesende Objekt und die Eigenschaft an.

In Dienst ReadRange in Parametergruppe Result(+):

Dieser Parameter des Typs BACnetObjectIdentifier gibt das Objekt an, das ausgelesen wurde.

In Dienst WriteProperty:

Dieser Parameter des Typs BACnetObjectIdentifier stellt eine Möglichkeit zur Bestimmung des Objektes dar, dessen Eigenschaft bearbeitet werden soll.

In Dienst WritePropertyMultiple im Parameter „Write Access Specification“:

Dieser Parameter des Typs BACnetObjectIdentifier bietet eine Möglichkeit zur Bestimmung des Objektes, dessen Eigenschaften bearbeitet werden sollen.

In Dienst Who-has:

Der Parameter „Object Identifier“ des Typs BACnetObjectIdentifier gibt den Object_Identifier des zu suchenden Objektes an. Ist der Parameter „Object Identifier“ nicht vorhanden, muss der Parameter Object Name vorhanden sein. Ist der Parameter „Object Identifier“ vorhanden, antworten nur Geräte mit einem Objekt, dessen Wert der Eigenschaft Object_Identifier mit dem Parameter „Object Identifier“ übereinstimmt und die wie in Kapitel „Gesamtliste der BACnet-Dienst-Parameter“, Parameter „Device Instance Range High Limit“ und „Device Instance Range Low Limit“ beschrieben für eine Antwort qualifiziert sind, mit einer I-Have-Dienstanfrage.

In Dienst I-Have:

Der Parameter „Object Identifier“ des Typs BACnetObjectIdentifier gibt den Object_Identifier des Objektes an, das als im initiiierenden Gerät enthalten angegeben wird. Dieser Identifier entspricht dem Wert der Eigenschaft Object_Identifier des angegebenen Objektes.

8.70 Object Name

Dienste I-Have, Who-Has

Beschreibung In Dienst Who-Has:

Der Parameter „Object Name“ des Typs CharacterString gibt den Wert der Eigenschaft Object_Name des zu suchenden Objektes an.

In Dienst I-Have:

Der Parameter „Object Name“ des Typs CharacterString gibt den Namen des Objektes an, das als im initiiierenden Gerät enthalten angegeben wird. Dieser Name entspricht dem Wert der Eigenschaft Object_Name des angegebenen Objektes.

8.71 Object Selection Criteria

Dienste ReadPropertyConditional

Beschreibung Dieser Parameter setzt sich aus dem Parameter „Selection Logic“ und unter Umständen einer Liste einer oder mehrerer „Selection Criteria“ zusammen. Jedes Kriterium besteht aus bis zu vier Parametern: „Property Identifier“, optionalem „Property Array Index“, „Relation Specifier“ und „Comparison Value“, siehe Kapitel „Parameter des Dienstes ReadPropertyConditional“ > „Object Selection Criteria“.

8.72 Password

Dienste	DeviceCommunicationControl, ReinitializeDevice
Beschreibung	Dieser optionale Parameter des Typs CharacterString gibt eine Zeichenkette mit maximal 20 Zeichen an. Bei Geräten mit Kennwortschutz wird der Dienst abgewiesen, falls dieser Parameter nicht vorhanden oder das Kennwort nicht korrekt ist. Geräte, die kein Kennwort erfordern, ignorieren diesen Parameter.

8.73 Priority

Dienste	ConfirmedEventNotification, GetEnrollmentSummary, UnconfirmedEventNotification, WriteProperty, referenziert durch WritePropertyMultiple im Parameter „Write Access Specification“
Beschreibung	In Dienst ConfirmedEventNotification, UnconfirmedEventNotification: Dieser Parameter des Typs Unsigned8 gibt die Priorität des aufgetretenen Ereignisses an. Die Priorität wird in der Eigenschaft Priority des Objektes Notification-Class angegeben. Es können Prioritäten zwischen „0“ und „255“ vergeben werden, eine niedrigere Zahl bedeutet eine höhere Priorität. Die Priorität und die Netzwerkpriorität werden wie in der Tabelle in Kapitel „Prioritätsklassifizierung von Alarmen und Ereignissen“ definiert zugewiesen. In Dienst GetEnrollmentSummary: Dieser Parameter des Typs Unsigned8 gibt die Priorität der von diesem Objekt generierten Benachrichtigungen an. In Dienst WriteProperty, WritePropertyMultiple im Parameter „Write Access Specification“: Dieser Parameter ist ein Integer im Bereich 1 bis 16, der die zugewiesene Priorität des Schreibvorgangs angibt. Beim Versuch, in eine Eigenschaft zu schreiben, die Befehle empfangen kann, ohne die Priorität anzugeben, wird standardmäßig eine Priorität von „16“ (niedrigste Priorität) angenommen. Beim Versuch, mit einer bestimmten Priorität in eine Eigenschaft zu schreiben, die keine Befehle empfangen kann, wird die Priorität ignoriert.

8.74 Priority Filter

Dienste GetEnrollmentSummary

Beschreibung Dieser Parameter besteht aus zwei Teilen, MinPriority und MaxPriority, des Datentyps Unsigned8. Er bietet eine Möglichkeit, nur die ereignisauslösenden Objekte zusammenzufassen, die Ereignisbenachrichtigungen mit einer in diesem Parameter angegebenen Priorität generieren können. Der Parameter „Priority Filter“ besteht aus zwei Teilen: MinPriority und MaxPriority. Alle ereignisauslösenden Objekte mit einer $\text{MinPriority} \leq \text{Priority} \leq \text{MaxPriority}$ sind in der Zusammenfassung enthalten. Ist der Parameter „Priority Filter“ nicht vorhanden, werden alle ereignisauslösenden Objekte ohne Berücksichtigung der Priorität zusammengefasst.

8.75 Process Identifier

Dienste ConfirmedEventNotification,
UnconfirmedEventNotification

Beschreibung Dieser Parameter des Typs Unsigned32 gibt den Prozess im empfangenden Gerät an, für den diese Benachrichtigung bestimmt ist.

8.76 Property Access Error

Dienste ConfirmedEventNotification, UnconfirmedEventNotification, referenziert durch ReadPropertyConditional und ReadPropertyMultiple in den Parametern „Object Selection Criteria“ und „List of Read Access Result“

Beschreibung Falls der Antwortende nicht auf die angegebene Eigenschaft des angegebenen Objektes zugreifen kann, wird dieser Parameter zurückgegeben. Er enthält einen Wert, der den Grund für das Fehlschlagen des Zugriffs angibt. Dieser Parameter besteht aus zwei Komponenten: einer Fehlerklasse und einem Fehlercode (siehe Kapitel „Fehler-, Ablehnungs- und Abbruchcodes“). Hierbei ist zu beachten, dass der Parameter nur das Fehlschlagen des Zugriffs auf eine bestimmte Eigenschaft eines bestimmten Objektes angibt. Der in Result(-) zurückgegebene Parameter „Error Type“ hingegen gibt das Fehlschlagen der gesamten ReadPropertyConditional- bzw. ReadPropertyMultiple-Dienstanfrage zurück.

8.77 Property Array Index

Dienste	ReadProperty, ReadRange, RemoveListElement, WriteProperty, referenziert durch ReadPropertyConditional in den Parametern „Object Selection Criteria“ und „Read Access Result“, referenziert durch ReadPropertyMultiple im Parameter „Read Access Result“, referenziert durch WritePropertyMultiple im Parameter „Write Access Specification“
Beschreibung	In Dienst ReadProperty, WriteProperty, WriteProperty-Multiple im Parameter „Write Access Specification“: Falls der Datentyp der oben angegebenen Eigenschaft Array ist, gibt dieser optionale Parameter des Typs Unsigned den Feldindex des Elementes der Eigenschaft an, auf die dieser Dienst verweist. Ist der Parameter „Property Array Index“ nicht vorhanden, bedeutet dies, dass auf das gesamte Feld verwiesen wird. Ist der Datentyp der oben angegebenen Eigenschaft nicht Array, ist dieser Parameter nicht vorhanden. In Dienst ReadRange in der Parametergruppe Argument: Falls der Datentyp der oben angegebenen Eigenschaft „Array of Lists“ ist, gibt dieser optionale Parameter des Typs Unsigned den Feldindex des Elementes der Eigenschaft an, auf die dieser Dienst verweist. Ist der Datentyp der oben angegebenen Eigenschaft nicht „Array of Lists“, ist dieser Parameter nicht vorhanden. Der Indexwert darf nicht null sein. In Dienst ReadRange in der Parametergruppe Result(+): Falls der Datentyp der oben angegebenen Eigenschaft „Array of Lists“ ist, gibt dieser Parameter des Typs Unsigned den Feldindex des Elementes der Eigenschaft an, auf die dieser Dienst verweist. Ist der Datentyp der oben angegebenen Eigenschaft nicht „Array of Lists“, ist dieser Parameter nicht vorhanden. In Dienst RemoveListElement, ReadPropertyMultiple im Parameter „Read Access Result“: Falls der Datentyp der oben angegebenen Eigenschaft Array ist, gibt dieser optionale Parameter des Typs Unsigned den Feldindex des Elementes der Eigenschaft an, die bearbeitet werden soll oder auf die dieser Dienst verweist. Andernfalls ist der Parameter nicht vorhanden. In Dienst ReadPropertyConditional im Parameter „Object Selection Criteria“: Falls der Datentyp der oben angegebenen Eigenschaft Array ist, gibt dieser optionale Parameter des Typs Unsigned den Feldindex des Elementes der Eigenschaft an, auf die dieser Dienst verweist. Andernfalls ist der Parameter nicht vorhanden. Ein „Property Array Index“-Wert von „Null“ ist nicht zulässig.

8.78 Property Identifier

Dienste	AddListElement, ReadProperty, ReadRange, RemoveListElement, WriteProperty, referenziert durch ReadPropertyConditional in den Parametern „Object Selection Criteria“ und „Read Access Result“, referenziert durch WritePropertyMultiple im Parameter „Write Access Specification“
Beschreibung	In Dienst AddListElement, RemoveListElement: Dieser Parameter des Typs BACnetPropertyIdentifier bietet eine Möglichkeit zur eindeutigen Bestimmung der Eigenschaft, die dieser Dienst bearbeitet. In Dienst ReadProperty in Parametergruppe Argument: Dieser Parameter des Typs BACnetPropertyIdentifier bietet eine Möglichkeit zur eindeutigen Bestimmung der Eigenschaft, die dieser Dienst ausliest und zurückgibt. Da der Dienst zum Auslesen einer einzelnen Eigenschaft eines einzelnen Objektes gedacht ist, darf der Wert dieses Parameters keiner der besonderen „Property Identifier“ ALL, REQUIRED oder OPTIONAL sein. In Dienst ReadProperty in Parametergruppe Result(+): Dieser Parameter des Typs BACnetPropertyIdentifier gibt die Eigenschaft an, die von diesem Dienst ausgelesen wurde und zurückgegeben wird. Da dieser Dienst zum Auslesen einer einzelnen Eigenschaft eines einzelnen Objektes gedacht ist, darf der Wert dieses Parameters keiner der besonderen „Property Identifier“ ALL, REQUIRED oder OPTIONAL sein. In Dienst ReadRange in Parametergruppe Argument: Dieser Parameter des Typs BACnetPropertyIdentifier gibt die von diesem Dienst auszulesende Eigenschaft an. Da dieser Dienst zum Auslesen einer einzelnen Eigenschaft eines einzelnen Objektes gedacht ist, darf der Wert dieses Parameters keiner der besonderen „Property Identifier“ ALL, REQUIRED oder OPTIONAL sein. In Dienst ReadRange in Parametergruppe Result(+): Dieser Parameter des Typs BACnetPropertyIdentifier gibt die ausgelesene Eigenschaft an.

In Dienst WriteProperty:

Dieser Parameter des Typs BACnetPropertyIdentifier bietet eine Möglichkeit zur eindeutigen Bestimmung der Eigenschaft, die dieser Dienst schreibt. Da dieser Dienst zum Schreiben einer einzelnen Eigenschaft eines einzelnen Objektes gedacht ist, darf der Wert dieses Parameters keiner der besonderen Property Identifier ALL, REQUIRED oder OPTIONAL sein.

In Parameter „Object Selection Criteria“:

Dieser Parameter des Typs BACnetPropertyIdentifier gibt die Eigenschaft an, deren Wert bei der Objektauswahl verwendet wird. Der Wert dieses Parameters darf keiner der besonderen „Property Identifier“ ALL, REQUIRED oder OPTIONAL sein.

In Parameter „Read Access Result“:

Dieser Parameter des Typs BACnetPropertyIdentifier gibt die Eigenschaft an, deren Wert zurückgegeben wird.

In Parameter „Write Access Specification“:

Dieser Parameter des Typs BACnetPropertyIdentifier bietet eine Möglichkeit zur eindeutigen Bestimmung der Eigenschaft, die dieser Dienst schreibt. Der Wert dieses Parameters darf keiner der besonderen „Property Identifier“ ALL, REQUIRED oder OPTIONAL sein.

8.79 Property Value

Dienste	ReadProperty, WriteProperty, referenziert durch ReadPropertyConditional im Parameter „Read Access Result“, referenziert durch ReadPropertyMultiple im Parameter „Read Access Result“, referenziert durch WritePropertyMultiple im Parameter „Write Access Specification“
Beschreibung	In Dienst ReadProperty: Bei erfolgreichem Zugriff auf die angegebene Eigenschaft des angegebenen Objektes wird dieser Parameter zurückgegeben. Er muss dem Datentyp der angegebenen Eigenschaft entsprechen und den Wert der angefragten Eigenschaft enthalten. In Dienst WriteProperty: Bei erfolgreichem Zugriff auf die angegebene Eigenschaft des angegebenen Objektes ersetzt dieser Parameter den Wert der Eigenschaft zum Zeitpunkt des Zugriffes. Der Parameter hat den für die Bearbeitung der entsprechenden Eigenschaft erforderlichen Datentyp.

In Dienst ReadPropertyConditional und**ReadPropertyMultiple im Parameter „Read Access Result:**

Bei erfolgreichem Zugriff auf die angegebene Eigenschaft des angegebenen Objektes wird dieser Parameter zurückgegeben. Er muss dem Datentyp der angegebenen Eigenschaft entsprechen und den Wert der angefragten Eigenschaft enthalten.

In Dienst WritePropertyMultiple im Parameter „Write Access Specification“:

Bei erfolgreichem Zugriff auf die angegebene Eigenschaft des angegebenen Objektes ersetzt dieser Parameter den Wert der Eigenschaft zum Zeitpunkt des Zugriffs. Der Parameter hat den für die Bearbeitung der entsprechenden Eigenschaft erforderlichen Datentyp.

8.80 Range

Dienste ReadRange

Beschreibung Dieser optionale Parameter gibt Kriterien für Elemente in einem zusammenhängenden Bereich der referenzierte Eigenschaft an, siehe Kapitel „Range“ „Vorgehensweise des Dienstes“. Der Parameter Range wird in der Tabelle des Kapitels „Range“ beschrieben. Verwendete Terminologie und Symbole werden in Kapitel „Terminologie der Darstellung“ erläutert.

8.81 Record Access

Dienste AtomicReadFile, AtomicWriteFile

Beschreibung **In Dienst AtomicReadFile in Parametergruppe Argument:** Der Parameter „Record Access“ gibt an, dass eintragsbasierter Dateizugriff erforderlich ist. „Record Access“ umfasst die Parameter „File Start Record“ und „Requested Record Count“.

In Parametergruppe Result(+):

Der Parameter „Record Access“ gibt an, dass eintragsbasierter Dateizugriff angefragt wurde. „Record Access“ umfasst die Parameter „File Start Record“, „Returned Record Count“ und „File Record Data“.

In Dienst AtomicWriteFile in Parametergruppe Argument:

Der Parameter „Record Access“ gibt an, dass eintragsbasierter Dateizugriff erforderlich ist. „Record Access“ umfasst die Parameter „File Start Record“, „Record Count“ und „File Record Data“.

In Parametergruppe Result(+):

Der Parameter „Record Access“ gibt an, dass eintragsbasierter Dateizugriff angefragt wurde. „Record Access“ umfasst den Parameter „File Start Record“. Der Parameter „File Start Record“ des Typs INTEGER gibt die Eintragsanzahl ab Dateianfang an, ab der die Daten geschrieben werden. Ein „File Start Record“ von „0“ ist der erste Eintrag der Datei.

8.82 Record Count

Dienste	AtomicWriteFile
Beschreibung	Dieser Parameter des Typs Unsigned gibt die Anzahl der Einträge der Datei an, die ab dem „File Start Record“ geschrieben werden.

8.83 Reference Index

Dienste	referenziert durch ReadRange im Parameter Range
Beschreibung	Der Parameter „Reference Index“ gibt den Index des ersten (falls Count positiv ist) oder des letzten (falls Count negativ ist) Elementes an, das gelesen wird.

8.84 Reference Sequence Number

Dienste	referenziert durch ReadRange im Parameter Range
Beschreibung	Der Parameter „Reference Sequence Number“ gibt die Sequenznummer des ersten (falls Count positiv ist) oder des letzten (falls Count negativ ist) Elementes an, das gelesen wird.

8.85 Reference Time

Dienste	referenziert durch ReadRange im Parameter Range
Beschreibung	Ist Count positiv, ist der erste gelesene Eintrag der Eintrag mit einem aktuelleren Zeitstempel als im Parameter „Reference Time“ angegeben. Ist Count negativ, ist der erste gelesene Eintrag der aktuellste Eintrag mit einem Zeitstempel älter als im Parameter „Reference Time“ angegeben.

8.86 Reinitialized State of Device

Dienste	ReinitializeDevice
Beschreibung	Dieser Parameter ermöglicht einem Client-BACnet-User, den gewünschten Gerätezustand nach der Neu-Initialisierung anzugeben. Der Wert des Parameters kann COLDSTART, WARMSTART, STARTBACKUP, ENDBACKUP, STARTRESTORE, ENDRESTORE oder ABORTRESTORE sein. WARMSTART bedeutet einen erneuten Versuch nach einem Neustart, hierbei werden alle Daten und Programme beibehalten, die auch bei einem kurzen Stromausfall erhalten bleiben.

8.87 Relation Specifier

Dienste	Referenziert durch ReadPropertyConditional im Parameter „Object Selection Criteria“												
Beschreibung	Dieser Parameter gibt einen von sechs booleschen Operatoren an: <table><tr><td>=</td><td>(gleich)</td><td>></td><td>(größer als)</td></tr><tr><td>≠</td><td>(ungleich)</td><td>≤</td><td>(kleiner oder gleich)</td></tr><tr><td><</td><td>(kleiner als)</td><td>≥</td><td>(größer oder gleich)</td></tr></table> Die booleschen Operatoren können nicht für jede Eigenschaft jedes Objektes verwendet werden. Bei einigen Eigenschaftswerten ist unter Umständen nur gleich (=) oder ungleich (≠) zulässig. Die folgende Tabelle zeigt, welche Operatoren mit welchem Datentyp verwendet werden können.	=	(gleich)	>	(größer als)	≠	(ungleich)	≤	(kleiner oder gleich)	<	(kleiner als)	≥	(größer oder gleich)
=	(gleich)	>	(größer als)										
≠	(ungleich)	≤	(kleiner oder gleich)										
<	(kleiner als)	≥	(größer oder gleich)										

Tabelle 77: Zulässige boolesche Operatoren für BACnet-Datentypen

Datentyp	Zulässige Operatoren	Datentyp	Zulässige Operatoren
NULL	$\neq$	CharacterString	$\neq < > \underline{\quad}$
BOOLEAN	$\neq$	OCTET STRING	$\neq$
Unsigned	$\neq < > \underline{\quad}$	ENUMERATED	$\neq$
INTEGER	$\neq < > \underline{\quad}$	Date	$\neq < > \underline{\quad}$
REAL	$\neq < > \underline{\quad}$	Time	$\neq < > \underline{\quad}$
BIT STRING	$\neq < >$	Sonstige	$\neq$
BACnet ObjectIdentifie	$\neq < > \underline{\quad}$		

8.88 Request

Dienste LifeSafetyOperation

Beschreibung Dieser Parameter des Typs BACnetLifeSafetyOperation gibt die angefragte Aktion folgendermaßen an:

{SILENCE, SILENCE_AUDIBLE, SILENCE_VISUAL,
RESET, RESET_ALARM, RESET_FAULT, UNSILENCE,
UNSILENCE_AUDIBLE, UNSILENCE_VISUAL}

8.89 Requested Octet Count

Dienste AtomicReadFile

Beschreibung Dieser Parameter des Typs Unsigned gibt die Anzahl der Oktette der Datei an, die ab der „File Start Position“ gelesen werden.

8.90 Requested Record Count

Dienste AtomicReadFile

Beschreibung Dieser Parameter des Typs Unsigned gibt die Anzahl der Einträge der Datei an, die ab dem „File Start Record“ gelesen werden.

8.91 Requesting Process Identifier

Dienste LifeSafetyOperation

Beschreibung Dieser Parameter des Typs Unsigned32 gibt einen Identifier mit Bedeutung für das sendende Gerät an, die den Prozess, der die Dienstanfrage sendet, eindeutig identifiziert. Zuweisung und Bedeutung der „Process Identifier“ sind lokal unterschiedlich.

8.92 Requesting Source

Dienste LifeSafetyOperation

Beschreibung Dieser Parameter des Typs CharacterString gibt die Identität des Anwenders an, der die LifeSafetyOperation-Dienstanfrage initiiert hat.

8.93 Result(-)

Dienste AcknowledgeAlarm, AtomicReadFile, AtomicWriteFile, ConfirmedCOVNotification, ConfirmedEventNotification, ConfirmedPrivateTransfer, ConfirmedTextMessage, CreateObject, DeleteObject, DeviceCommunicationControl, GetAlarmSummary, GetEnrollmentSummary, GetEventInformation, LifeSafetyOperation, Object Access, ReadProperty, ReadPropertyConditional, ReadPropertyMultiple, ReadRange, ReinitializeDevice, RemoveListElement, SubscribeCOV, SubscribeCOVProperty, VT-Close, VT-Data, VT-Open, WriteProperty, WritePropertyMultiple

Beschreibung Der Parameter Result(-) gibt an, dass die Dienstanfrage fehlgeschlagen ist. Der Grund hierfür ist im Parameter Error Type angegeben.

In Dienst AddListElement:

Keines der angegebenen Elemente wurde der Liste hinzugefügt.

In Dienst RemoveListElement:

Keines der Elemente des angegebenen Objektes wird entfernt.

In Dienst DeleteObject:

Das angegebene Objekt wurde nicht gelöscht.

In Dienst WritePropertyMultiple:

Der Parameter Result(-) gibt an, dass mindestens eine der angegebenen Eigenschaften nicht wie angefragt geändert werden konnte. Der Grund hierfür wird im Parameter „Error Type“ zusammen mit „Object Identifier“, „Property Identifier“ und „Property Array Index“ der ersten angegebenen Eigenschaften angegeben, die nicht korrekt geschrieben werden konnte.

8.94 Result(+)

Dienste AcknowledgeAlarm, AtomicReadFile, AtomicWriteFile, ConfirmedCOVNotification, ConfirmedEventNotification, DeviceCommunicationControl, GetAlarmSummary, GetEnrollmentSummary, GetEventInformation, LifeSafetyOperation, ObjectAccess, ReadProperty, ReadPropertyConditional, ReadPropertyMultiple, ReadRange, ReinitializeDevice, RemoveListElement, SubscribeCOV, SubscribeCOVProperty, WriteProperty, WritePropertyMultiple

Beschreibung Der Parameter Result(+) gibt an, dass die Dienstanfrage erfolgreich war. Dies umfasst unter anderem folgende Parameter:

In Dienst AddListElement:

Alle der angegebenen Elemente wurden der Liste hinzugefügt.

In Dienst RemoveListElement:

Alle der angegebenen Elemente wurden entfernt.

8.95 Result Flags

Dienste ReadRange

Beschreibung Dieser Parameter des Typs BACnetResultFlags gibt mehrere Flags an, die Merkmale der zurückgegebenen Daten beschreiben:

{FIRST_ITEM, LAST_ITEM, MORE_ITEMS}

Das Flag FIRST_ITEM gibt an, ab die Antwort das erste Element der Liste oder des Feldes (bei einem Positionsindex) oder das Element mit dem ältesten Zeitstempel (bei einem chronologischen Index) enthält.

Das Flag LAST_ITEM gibt an, ab die Antwort das letzte Element der Liste oder des Feldes (bei einem Positionsindex) oder das Element mit dem aktuellsten Zeitstempel (bei einem chronologischen Index) enthält.

Das Flag MORE_ITEMS gibt an, ob mehrere Elemente der Anfrage entsprachen, jedoch nicht in der APDU übertragen werden konnten.

8.96 Returned Record Count

Dienste AtomicReadFile

Beschreibung Dieser Parameter des Typs Unsigned gibt die Anzahl der Einträge an, die tatsächlich aus der Datei ausgelesen wurden, dieser Wert kann geringer als der „Requested Record Count“ sein.

8.97 Segmentation Supported

Dienste I-Am

Beschreibung Dieser Parameter des Typs BACnetSegmentation gibt die Fähigkeit des Gerätes zur Initiierung von I-Am-Dienstanfragen bezüglich der Verarbeitung segmentierter Nachrichten an. Der Wert dieses Parameters entspricht dem der Eigenschaft Segmentation_Supported des Device-Objektes.

8.98 Selection Logic

Dienste	Referenziert durch ReadPropertyConditional im Parameter „Object Selection Criteria“
Beschreibung	Dieser Parameter gibt den Umfang der Objekte an, deren Eigenschaften zurückgegeben werden. Es sind drei Werte zulässig: AND, OR und ALL. Wird AND oder OR verwendet, gibt der Parameter die Methode für die Kombination mehrerer Auswahlkriterien zur Bestimmung an, ob ein bestimmtes Objekt zu den Objekten gehört, deren Eigenschaften zurückgegeben werden. Bei AND müssen alle Auswahlkriterien für das Objekt TRUE sein, bei OR muss mindestens ein Kriterium für das Objekt TRUE sein. Bei ALL werden alle Objekte in der Datenbank des antwortenden BACnet-Users ausgewählt und die „List of Selection Criteria“ ist in der Anfrage nicht enthalten.

8.99 Stream Access

Dienste	AtomicReadFile, AtomicWriteFile
Beschreibung	In Dienst AtomicWriteFile: Der Parameter „Stream Access“ gibt an, ob stream-basierter Dateizugriff erforderlich ist. „Stream Access“ umfasst die Parameter „File Start Position“ und „Requested Octet Count“. In Dienst AtomicReadFile: Der Parameter „Stream Access“ gibt an, ob stream-basierter Dateizugriff angefordert wurde. „Stream Access“ umfasst die Parameter „File Start Position“ und „File Data“.

8.100 Subscriber Process Identifier

Dienste	ConfirmedCOVNotification, SubscribeCOV, SubscribeCOVProperty, UnconfirmedCOVNotification
Beschreibung	In Dienst ConfirmedCOVNotification: Dieser Parameter des Typs Unsigned32 gibt einen numerischen Handle mit Bedeutung für den Abonnenten an. Der Handle wird für die Bestimmung des Prozesses im COV-Client verwendet, der die Benachrichtigung erhalten soll. In Dienst UnconfirmedCOVNotification: Dieser Parameter des Typs Unsigned32 gibt einen numerischen Handle mit Bedeutung für den Abonnenten an. Der Handle wird für die Bestimmung des Prozesses im COV-Client verwendet, der die Benachrichtigung erhalten soll. Der Wert „Null“ ist für UnsubscribedCOV reserviert. In Dienst SubscribeCOV-, SubscribeCOVProperty: Dieser Parameter des Typs Unsigned32 gibt einen numerischen Handle mit Bedeutung für den Abonnenten an. Der Handle wird verwendet, um künftige Neu-Abonnements und Stornierungen des Abonnenten mit dem COV-Kontext im COV-Servergerät abzugleichen und um bei zu bestätigenden oder nicht zu bestätigenden COV-Benachrichtigungen den Prozess im COV-Client zu bestimmen, der diese empfangen soll. Der Wert „Null“ ist für UnsubscribedCOV-Benachrichtigungen reserviert, siehe Kapitel „UnconfirmedCOVNotification-Dienst“.

8.101 Text Message Source Device

Dienste	ConfirmedTextMessage, UnconfirmedTextMessage
Beschreibung	Dieser Parameter des Typs BACnetObjectIdentifier gibt den Wert der Eigenschaft Object_Identifier des Device-Objektes an, das diese Textnachricht initiiert hat.

8.102 Time

Dienste	TimeSynchronization, UTCTimeSynchronization
Beschreibung	Dieser Parameter des Typs BACnetDateTime gibt die aktuelle Uhrzeit und das Datum an. Im Dienst TimeSynchronization wird durch die Uhr im Gerät bestimmt, das die Dienstanfrage initiiert.

8.103 Time Duration

Dienste DeviceCommunicationControl

Beschreibung Dieser optionale Parameter des Typs Unsigned16 gibt den Zeitraum in Minuten an, den das entfernte Gerät alle APDUs außer DeviceCommunicationControl und, falls unterstützt, ReinitializeDevice ignoriert. Ist der Parameter „Time Duration“ nicht vorhanden, wird ein unbegrenzter Zeitraum angenommen, d. h. nur die APDUs DeviceCommunicationControl oder ReinitializeDevice aktivieren die Kommunikation. Der Parameter „Time Duration“ wird ignoriert und ein unendlicher Zeitraum angenommen, wenn der Parameter Enable/Disable den Wert „ENABLE“ hat.

8.104 Time Of Acknowledgment

Dienste AcknowledgeAlarm

Beschreibung Dieser Parameter des Typs BACnetTimeStamp gibt die Zeit an, zu welcher der Anwender oder Prozess die Ereignisbenachrichtigung bestätigt hat.

8.105 Time Remaining

Dienste ConfirmedCOVNotification, UnconfirmedCOVNotification

Beschreibung Dieser Parameter des Typs Unsigned gibt die verbleibende Lebensdauer für das Abonnement in Sekunden an. Ein Wert von „Null“ gibt an, dass das Abonnement unbegrenzt gültig ist und keine automatische Stornierung erfolgt.

In Dienst UnconfirmedCOVNotification:

Ein Wert von „Null“ kann ebenso eine Benachrichtigung ohne Abonnement anzeigen.

8.106 Time Stamp

Dienste	AcknowledgeAlarm, ConfirmedEventNotification, UnconfirmedEventNotification
Beschreibung	In Dienst ConfirmedEventNotification, UnconfirmedEventNotification: Dieser Parameter des Typs BACnetTimeStamp gibt die aktuelle Zeit laut Geräteuhr an, welche die Dienstanfrage initiiert. Hat das Gerät keine Uhr, gibt dieser Parameter eine Sequenznummer an, welche die relative Reihenfolge dieser Ereignisbenachrichtigung im Vergleich mit anderen Ereignisbenachrichtigungen dieses Gerätes unabhängig vom Empfänger angibt. Die Sequenznummer steigt stetig und kann auch mittels Modulo-Arithmetik implementiert werden. Ein Gerät kann über eine einzelne Sequenznummer für alle ereignisauslösenden Objekte verfügen oder es kann für jedes Objekt eine eigene Sequenznummer verwendet werden. In Dienst AcknowledgeAlarm: Dieser Parameter des Typs BACnetTimeStamp gibt denselben „Time Stamp“ an, der in der durch diesen Dienst bestätigten Ereignisbenachrichtigung enthalten war. Der „Time Stamp“ wird vom Empfänger dieses Dienstes verwendet, um die bestätigte Ereignisbenachrichtigung zu identifizieren, falls mehr als eine Benachrichtigung mit demselben „To State“ ausgegeben wurde.

8.107 To State

Dienste	ConfirmedEventNotification, UnconfirmedEventNotification
Beschreibung	Dieser Parameter des Typs BACnetEventState gibt den Wert der Eigenschaft Event_State des Objektes nach Auftreten des Ereignisses an, das diese Benachrichtigung initiiert hat.

8.108 Vendor ID

Dienste	ConfirmedPrivateTransfer, UnconfirmedPrivateTransfer
Beschreibung	Dieser Parameter des Typs Unsigned gibt den eindeutigen ID-Code des Herstellers für den Typ des auszuführenden herstellerspezifischen Dienstes an.

8.109 Vendor Identifier

Dienste I-Am

Beschreibung Dieser Parameter des Typs Unsigned16 gibt die Identität des Geräteherstellers mit einer I-Am-Dienstanfrage an.

Abbildungsverzeichnis

Abbildung 1: Beispiel eines Accumulator-Objektes	19
Abbildung 2: Hasse-Diagramm.....	32
Abbildung 3: Hasse-Diagramm (Silenced)	33
Abbildung 4: Hasse-Diagramm (Operation_Expected)	33
Abbildung 5: Verhältnis zwischen Pulse-Converter- und Accumulator-Objekt...	38
Abbildung 6: Beispiel für Event-Enrollment	55
Abbildung 7: Loop-Objektstruktur mit referenzierten Objekten	182

Tabellenverzeichnis

Tabelle 1: Darstellungen der Zahlensysteme	13
Tabelle 2: Schriftkonventionen	13
Tabelle 3: Eigenschaften des Accumulator-Objektes	20
Tabelle 4: Eigenschaften des Analog-Value-Objektes.....	21
Tabelle 5: Eigenschaften des Averaging-Objektes	22
Tabelle 6: Eigenschaften des Binary-Value-Objektes.....	23
Tabelle 7: Eigenschaften des Calendar-Objektes	24
Tabelle 8: Eigenschaften des Command-Objektes.....	26
Tabelle 9: Eigenschaften des File-Objektes	27
Tabelle 10: Eigenschaften des Group-Objektes	28
Tabelle 11: Present_Value des Life-Safety-Point-Objektes.....	30
Tabelle 12: Eigenschaften des Life-Safety-Point-Objektes	31
Tabelle 13: Eigenschaften des Life-Safety-Zone-Objektes.....	34
Tabelle 14: Eigenschaften des Multi-State-Value-Objektes	35
Tabelle 15: Eigenschaften des Notification-Class-Objektes	37
Tabelle 16: Eigenschaften des Pulse-Converter-Objektes	39
Tabelle 17: Eigenschaften des Schedule-Objektes.....	40
Tabelle 18: Eigenschaften des Trend-Log-Objektes	43
Tabelle 19: Standardisierte Objekte, die unter Umständen COV-Reporting unterstützen	48
Tabelle 20: Kriterien für die Verwendung von COV-Reporting mit anderen Eigenschaften als in voriger Tabelle aufgeführt	48
Tabelle 21: Standardisierte Objekte, die objektinternes Melden unterstützen	50
Tabelle 22: : Standard-Objekteigenschaftswerte, die in Benachrichtigungen zurückgegeben werden.....	52
Tabelle 23: Benachrichtigungsparameter für Standard-Ereignistypen	52
Tabelle 24: Alarm- und Ereignispriorität - Zuweisung von Netzwerkpriorität.....	56
Tabelle 25: Struktur der Funktionen des Dienstes AcknowledgeAlarm	59
Tabelle 26: Struktur der Funktionen des Dienstes ConfirmedCOVNotification ..	60
Tabelle 27: Struktur der Funktionen des Dienstes UnconfirmedCOVNotification	61
Tabelle 28: Struktur der Funktionen des Dienstes ConfirmedEventNotification .	62
Tabelle 29: Struktur der Funktionen des Dienstes UnconfirmedEventNotification	63
Tabelle 30: Struktur der Funktionen des Dienstes GetAlarmSummary	64
Tabelle 31: Struktur der Funktionen des Dienstes GetEnrollmentSummary	65
Tabelle 32: Struktur der Funktionen des Dienstes GetEventInformation	66
Tabelle 33: Struktur der Funktionen des Dienstes LifeSafetyOperation	68
Tabelle 34: Struktur der Funktionen des Dienstes SubscribeCOV	69
Tabelle 35: Struktur der Funktionen des Dienstes SubscribeCOVProperty	71
Tabelle 36: Struktur der Funktionen des Dienstes AtomicReadFile	74
Tabelle 37: Struktur der Funktionen des Dienstes AtomicWriteFile	75
Tabelle 38: Struktur der Funktionen des Dienstes AddListElement	76
Tabelle 39: Struktur der Funktionen des Dienstes RemoveListElement	77
Tabelle 40: Struktur der Funktionen des Dienstes ReadProperty	78
Tabelle 41: Struktur der Funktionen des Dienstes ReadPropertyConditional	79
Tabelle 42: Struktur des Parameters „Object Selection Criteria“	80

Tabelle 43: Struktur des Parameters „Read Access Result“	80
Tabelle 44: Struktur der Funktionen des Dienstes ReadPropertyMultiple.....	81
Tabelle 45: Struktur des Parameters Read Access Specification	82
Tabelle 46: Struktur des Parameters „Read Access Result“	82
Tabelle 47: Struktur der Funktionen des Dienstes ReadRange	83
Tabelle 48: Struktur des Parameters Range	83
Tabelle 49: Struktur der Funktionen des Dienstes WriteProperty	86
Tabelle 50: Struktur der Funktionen des Dienstes WritePropertyMultiple.....	87
Tabelle 51: Struktur des Parameters „Write Access Specification“	88
Tabelle 52: Struktur der Funktionen des Dienstes DeviceCommunicationControl	89
Tabelle 53: Struktur der Funktionen des Dienstes ReinitializeDevice.....	90
Tabelle 54: Struktur der Funktionen des Dienstes TimeSynchronization.....	91
Tabelle 55: Struktur der Funktionen des Dienstes UTCTimeSynchronization....	92
Tabelle 56: Struktur der Funktionen des Dienstes Who-Has	93
Tabelle 57: Struktur der Funktionen des Dienstes I-Have	94
Tabelle 58: Struktur der Funktionen des Dienstes Who-Is	95
Tabelle 59: Struktur der Funktionen des Dienstes I-Am.....	96
Tabelle 60: Fehlerklasse - DEVICE	97
Tabelle 61: Fehlerklasse - OBJECT	98
Tabelle 62: Fehlerklasse - PROPERTY	99
Tabelle 63: Fehlerklasse - RESOURCES	100
Tabelle 64: Fehlerklasse - SECURITY	101
Tabelle 65: Fehlerklasse - SERVICES.....	102
Tabelle 66: Fehlerklasse - VT	103
Tabelle 67: Ablehnungsgründe	104
Tabelle 68: Abbruchgründe.....	105
Tabelle 69: Komponenten und Datentypen der BACnetActionCommand	113
Tabelle 70: Komponenten einer BACnetDestination.....	171
Tabelle 71: Optionen des Typs BACnetScale	177
Tabelle 72: Vergleichswerte zur Auswahl von Eigenschaften.....	199
Tabelle 73: Fehlerklasse und Fehlercode des Dienstes CreateObject.....	205
Tabelle 74: Fehlerklasse und Fehlercode des Dienstes DeleteObject.....	205
Tabelle 75: Fehlerklasse und Fehlercode der Dienste DeleteObject ReadProperty/ReadPropertyMultiple.....	205
Tabelle 76: Fehlerklasse und Fehlercode der Dienste WriteProperty/WritePropertyMultiple	206
Tabelle 77: Zulässige boolesche Operatoren für BACnet-Datentypen	235

WAGO Kontakttechnik GmbH & Co. KG
Postfach 2880 • D-32385 Minden
Hansastraße 27 • D-32423 Minden
Telefon: 05 71/8 87 – 0
Telefax: 05 71/8 87 – 1 69
E-Mail: info@wago.com
Internet: <http://www.wago.com>

